2011

A Triangulation-Based Approach to Nonrigid Image Registration

Timothy R. Linden
Wright State University

A Triangulation-Based Approach to Nonrigid Image Registration

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering

By

Timothy R Linden
B.S.C.E, Cedarville University, 2006

2011

Wright State University

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

May 5, 2011

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Timothy Linden ENTITLED A Triangulation-Based Approach to Nonrigid Image Registration ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Masters of Science in Computer Engineering

Arthur Goshtasby, Ph. D.
Thesis Director

Mateen Rizki, Ph. D.
Department Chair

Committee on
Final Examination

Arthur Goshtasby, Ph. D.

Thomas Wischgoll, Ph. D.

Jack Jean, Ph. D.

Andrew T. Hsu, Ph. D.
Dean, School of Graduate Studies

ABSTRACT

Linden, Timothy. M.S.C.E., Department of Computer Science and Engineering, Wright State University, 2011. A Triangulation-Based Approach to Nonrigid Image Registration

A triangulation-based approach to nonrigid image registration is presented. This method builds upon control point projective registration methods. Control points for this method are located using the Harris point detector. An analysis is presented for this detector to show its properties. Projective registration is used as the basis for this non-rigid registration method. Details of the projective registration method used are presented. Nonrigid registration is used to spatially align images of a 3-D scene taken from different views. Projective registration approximates the scene geometry as a plane. This nonrigid approach subdivides the images into small corresponding triangles, to improve the approximation of the scene geometry. Affine transformation functions are used to register corresponding triangles. Finally a refinement step is presented to smooth the transition between adjacent triangles and achieve a smooth registration across the image domain.

TABLE OF CONTENTS

Chapter 1 Introduction and Background 1

1.1 Image Registration 1

1.2 Feature Point Detection 2

1.2.1 Harris Corner Detector 3

1.2.2 Laplacian of Gaussian 8

1.2.3 Control Point Detection Performance 10

1.3 Registration 11

1.3.1 Projective Registration 12

1.3.2 Piecewise Linear Image Registration 12

1.3.3 Weighted Linear Image Registration 13

Chapter 2 Analysis of Harris and LoG Point Detectors 14

2.1 Harris Point Detection 14

2.1.1 Performance Analysis 19

2.1.2 Algorithm 19

2.1.3 Results 21

2.1.4 Data Analysis 21

2.2 Laplacian of Gaussian Point Detection 24

2.2.1 Algorithm 25

2.2.2 Results 25

Chapter 3 Projective Image Registration 28

3.1 Control Points 30

3.2 Template Matching 31
3.3 RANSAC Matching 34
3.4 Projective Transformation 36
3.5 Projective Registration Results 39
Chapter 4 Piecewise Linear Registration 40
4.1 Steps for piecewise linear registration 41
4.2 Scale Reduction 41
4.3 Tracking corresponding points 42
4.3.1 Algorithm for tracking the points 43
4.3.2 Tracking corresponding points with increasing scale 44
4.4 Triangulation of the points 45
4.5 Register Corresponding Triangles 47
4.6 Piecewise Linear Registration Results 50
Chapter 5 Weighted Linear Registration 51
5.1 Rational Gaussian Basis Functions 51
5.2 Weighted Linear Registration Results 55
Chapter 6 Results 58
6.1 Pentagon Images 59
6.2 Mars Images 62
6.3 Aerial Images 64
6.4 City Images 66
6.5 Cube Images 68
6.6 Crater Images 70

6.7 OSU Images 72

Chapter 7 Conclusion 74

7.1 Registration 74

7.2 Control-point detection 75

7.3 Piecewise linear registration 75

7.4 Weighted linear registration 76

Chapter 8 Future Work 78

Bibliography 80

LIST OF FIGURES

Figure Page

Figure 2.1 : Mars1.pgm 21
Figure 2.2 : Pentagon.pgm 21
Figure 2.3 : Mars Number of Corresponding Points 21
Figure 2.4 : Pentagon Number of Corresponding Points 21
Figure 2.5 : Mars Localization Error.....22
Figure 2.6 : Pentagon Localization Error 22
Figure 2.7 : Mars Repeatability.....23
Figure 2.8 : Pentagon Repeatability 23
Figure 2.9 : Mars Number of Corner Points.....24
Figure 2.10 : Pentagon Number of Corner Points.....24
Figure 2.11 : Mars LoG Number of Corresponding Points.....26
Figure 2.12 : Pentagon LoG Number of Corresponding Points 26
Figure 2.13 : Mars LoG Localization Error.....26
Figure 2.14 : Pentagon LoG Localization Error 26
Figure 2.15 : Mars LoG Repeatability.....27
Figure 2.16 : Pentagon LoG Repeatability.....27
Figure 3.1 : pentagon1.pgm 31
Figure 3.2 : corner map for pentagon1.pgm 31
Figure 3.3 : Pentagon 1 33
Figure 3.4 : 15 x 15 pixel template from Pentagon 1 33

Figure 3.5 : Template response on Pentagon 2 33

Figure 3.6 : Pentagon 2 33

Figure 3.7 : Data set with gross error 35

Figure 3.8 : Data set with error removed 35

Figure 3.9 : $\boldsymbol{f_x(x,y)}$ 38

Figure 3.10 : $\boldsymbol{f_y(x,y)}$ 38

Figure 3.11 : Registered Pentagon images 39

Figure 3.12 : Difference of the registered images 39

Figure 4.1 : Pentagon 1 image at 1/4 scale 44

Figure 4.2 : Pentagon 2 image at 1/4 scale 44

Figure 4.3 : Pentagon 1 image at 1/2 scale 44

Figure 4.4 : Pentagon 2 image at 1/2 scale 44

Figure 4.5 : Pentagon 1 image at full scale 44

Figure 4.6 : Pentagon 2 image at full scale 44

Figure 4.7 : Invalid Triangulation 46

Figure 4.8 : Edge flipping to create valid triangulation 46

Figure 4.9 : Triangulation for Pentagon 1 image 47

Figure 4.10 : Triangulation for Pentagon 2 image 47

Figure 4.11 : $\boldsymbol{f_x(x,y)}$ 49

Figure 4.12 : $\boldsymbol{f_y(x,y)}$ 49

Figure 4.13 : Piecewise linear registration: Image overlaying 50

Figure 4.14 : Piecewise linear registration: Absolute intensity differences ...50

Figure 5.1 : Circumcircle of a thin triangle re-centered to the centroid of the triangle 54
Figure 5.2 : Circle created from the minimum vertex distance centered at the centroid of the triangle.....55
Figure 5.3 : Weighted linear overlay result 56
Figure 5.4 : Weighted linear absolute differences.....56
Figure 5.5 : $\boldsymbol{X}(\boldsymbol{x},\boldsymbol{y})$ map 57
Figure 5.6 : $\boldsymbol{Y}(\boldsymbol{x},\boldsymbol{y})$ map.....57
Figure 6.1 : Pentagon 1 image 59
Figure 6.2 : Pentagon 2 image 59
Figure 6.3 : Projective registration: Overlaying result 60
Figure 6.4 : Projective registration: Absolute difference result.....60
Figure 6.5 : Piecewise linear registration: Overlaying result 60
Figure 6.6 : Piecewise linear registration: Absolute difference result.....60
Figure 6.7 : Weighted linear registration: Overlaying result 61
Figure 6.8 : Weighted linear registration: Absolute difference result.....61
Figure 6.9: Mars 1 62
Figure 6.10 : Mars 2 62
Figure 6.11 : Projective registration : Overlaying result 63
Figure 6.12 : Projective registration : Absolute difference result.....63
Figure 6.13 : Piecewise linear registration : Overlaying result 63
Figure 6.14 : Piecewise linear registration : Absolute difference result.....63
Figure 6.15 : Weighted linear registration : Overlaying result 63

Figure 6.16 : Weighted linear registration : Absolute difference result.....63
Figure 6.17 : Aerial 1 image 64
Figure 6.18 : Aerial 2 image 64
Figure 6.19 : Projective overlaying result.....64
Figure 6.20 : Projective difference result 64
Figure 6.21 : Piecewise linear overlaying result 65
Figure 6.22 : Piecewise linear difference result 65
Figure 6.23 : Weighted linear overlaying result 65
Figure 6.24 : Weighted linear difference result 65
Figure 6.25 : City 1 image 66
Figure 6.26 : City 2 image 66
Figure 6.27 : Projective overlaying result.....66
Figure 6.28 : Projective difference result 66
Figure 6.29 : Piecewise linear overlaying result 67
Figure 6.30 : Piecewise linear difference result 67
Figure 6.31 : Weighted linear overlaying result 67
Figure 6.32 : Weighted linear difference result 67
Figure 6.33 : Cube 1 image 68
Figure 6.34 : Cube 2 image 68
Figure 6.35 : Projective overlaying result.....68
Figure 6.36 : Projective difference result 68
Figure 6.37 : Piecewise linear overlaying result 69
Figure 6.38 : Piecewise linear difference result 69

Figure 6.39 : Weighted linear overlaying result 69
Figure 6.40 : Weighted linear difference result 69
Figure 6.41 : Crater 1 image.....70
Figure 6.42 : Crater 2 image.....70
Figure 6.43 : Projective overlaying result.....70
Figure 6.44 : Projective difference result 70
Figure 6.45 : Piecewise linear overlaying result 71
Figure 6.46 : Piecewise linear difference result 71
Figure 6.47 : Weighted linear overlaying result 71
Figure 6.48 : Weighted linear difference result 71
Figure 6.49 : OSU 1 image.....72
Figure 6.50 : OSU 2 image.....72
Figure 6.51 : Projective overlaying result.....72
Figure 6.52 : Projective difference result 72
Figure 6.53 : Piecewise linear overlaying result 73
Figure 6.54 : Piecewise linear difference result 73
Figure 6.55 : Weighted linear overlaying result 73
Figure 6.56 : Weighted linear difference result 73

Chapter 1
Introduction and Background

1.1 Image Registration

Image registration is the process of spatially aligning two or more images of a scene. This is a fundamental capability required in computer vision, allowing variations in a scene to be compared. Applications of image registration are target tracking, panoramic image stitching, medical image fusion, satellite image fusion, robotic and computer vision applications for depth perception [1]. Target tracking requires image registration to compensate for the motion of the imaging sensor so that the motion of the targets can be properly estimated. Image stitching is a well-studied application of image registration. This aligns and corrects images to generate the final panoramic view through image registration. Image registration is an important step in medical image analysis. Images are taken that represent the internal body structures and need to be compared to previously captured images of the same structure to determine variations for diagnosis. Remote sensing applications may require that a geospatial area of the earth be monitored for changes over time. Vegetation changes and land cover changes can show how geological systems can affect surrounding areas or how city growth can affect the environment.

A typical image registration algorithm has the following steps. [1]

- Feature Detection. Features are points, lines, or other image structures that are manually selected or automatically detected.
- Feature Matching. The process of finding correspondence between detected features.
- Transformation model estimation. The mapping function to align the images is estimated from the feature correspondences.
- Image resampling. One image is resampled using the transformation model to align it with the other image.

1.2 Feature Point Detection

There exist many methods for locating points in an image. In this study only the Harris [2] corner detector and the Laplacian of Gaussian [3] (LoG) blob detector are used. Control points detected in the images are matched using the random sample and consensus (RANSAC) [4] method to determine point correspondences. These correspondences are then used to determine a transformation to align the images. A transformation function defines how points in one image are related to the points in another image.

Corner detection is widely used in machine vision to allow a robot to navigate in its environment; Moravec [8] developed one of the earliest corner detectors. However, this detector showed significant flaws due to only using information at 45 degree increments, making it anisotropic. This causes the

operator to respond too readily along certain edges. Harris and Stephens [2] modified this corner detector to build a new detector that is based on auto-correlation to resolve the anisotropic problem of the Moravec operator. The resulting operator has been shown to have significantly better results, locating corners more accurately.

Laplacian of Gaussian filters have been found to be very effective in finding blob like structures in an image. Lindeberg reviews Scale-space theory [5] and shows that Gaussian and its derivatives are the only possible smoothing kernels, under some general assumptions, that can be used in scale-space analysis. For this reason Lowe used it to develop the popular Scale Invariant Feature Transform (SIFT) [6] point detector. The results presented by Lowe demonstrate the robustness of SIFT under scale and rotation and the ability of it to locate reliable feature points. In order to optimize SIFT Lowe used the Difference of Gaussian (DoG) approximation, instead of the Laplacian of Gaussian, as suggested by D. Marr [7] in an edge detection study.

1.2.1 Harris Corner Detector

Harris and Stephens [2] began with the Moravec operator to develop a more reliable detector. Moravec developed a detector to find interesting points by looking at discrete directions of a window of size 4 or 8 pixels on each side. This resulted in a detector which is not rotationally invariant. To resolve this issue Harris and Stephens setup an inertia matrix. Mathematically this operator is rotationally invariant; however, the method by which the image gradient is

computed can affect the rotational invariance property [9]. Since images are discrete approximations of continuous images, a computer can only make discrete approximations when calculating the gradient of the image. This results in degradation of the isotropic behavior of the corner detector. By using a more accurate Gaussian kernel the effect of the discrete gradient operations can be reduced [9].

A corner detector is naturally defined as a location where there is a gradient in two directions orthogonal to each other. In a two dimensional image this may look like the corner of a box or window. The strength of a corner is defined by the strength of the gradient in each direction and how orthogonal the gradients are. This is similar to finding the magnitude of the cross product of two vectors representing the gradient in each direction. A larger magnitude of the cross product indicates the magnitudes of both vectors are large. The magnitude of the cross product is the determinate of the matrix representing the vectors, such as an inertia matrix.

Harris and Stephens developed the corner detector using the following:

$$E_{x,y} = \sum_{u,v} \left| I_{x+u,y+v} - I_{u,v} \right|^2 w_{u,v} \tag{1.1}$$

A patch in image I is defined by a rectangular window $w_{u,v}$ which assumes value one in the window and zero everywhere else. To find a corner, the patch is shifted by (x, y) and the sum of the square difference is calculated. In this equation $E_{x,y}$ is the change produced by shifting the patch by (x, y) and

comparing it to itself. When the difference is small, there is little change in intensity in the window. If the difference is large, there is a large change in the intensity. The minimum value of the shifts, $\{(1,0),(1,1),(0,1),(-1,1)\}$, is used to determine the corner strength for the patch. Moravec's detector is the local maximum of the minimum of $\{E\}$:

$$E_{x,y} = \sum_{u,v} w_{u,v}\left[xI_x + yI_y + O(x^2, y^2)\right]^2 \tag{1.2}$$

The function can be rewritten in such a way that the shift is with respect to the origin of the patch. In this case the (x, y) terms are multiplied by the gradient in the X and Y directions. For small shifts, this is the sum of the square difference centered at (x, y) under the window $w_{u,v}$. Approximating the gradients for such small shifts can be written as,

$$I_x = I \otimes (-1,0,1) \approx \frac{\partial I}{\partial x}, \tag{1.3}$$

$$I_y = I \otimes (-1,0,1)^T \approx \frac{\partial I}{\partial y}. \tag{1.4}$$

Since only small shifts are being considered, the function E can be rewritten as,

$$E = Ax^2 + 2Cxy + By^2, \tag{1.5}$$

$$A = {I_x}^2 \otimes w, \tag{1.6}$$

$$B = {I_y}^2 \otimes w, \tag{1.7}$$

$$C = I_x I_y \otimes w. \tag{1.8}$$

To smooth the response, window $w_{u,v}$ can be replaced by a Gaussian,

$$w = e^{-\frac{u^2+v^2}{2\sigma^2}} \tag{1.9}$$

This equation can be rewritten using an inertia matrix to simplify the analysis and determine corners. An inertia matrix describes the variation of a mass about a point, in the same way the inertia matrix here will describe the variation of the intensities about a point in two dimensions.

$$E(x,y) = \begin{bmatrix} x \\ y \end{bmatrix} M \begin{bmatrix} x & y \end{bmatrix} \tag{1.10}$$

The matrix M is defined by the terms A, B and C from equations (1.6), (1.7) and (1.8):

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \tag{1.11}$$

Replacing terms A, B and C as they are defined above gives,

$$M = \begin{bmatrix} \overline{I_x^2} & \overline{I_x I_y} \\ \overline{I_x I_y} & \overline{I_y^2} \end{bmatrix} \tag{1.12}$$

which is the more recognizable form of the inertia matrix.

Corners of an object are defined by the gradients, therefore; the inertia matrix is the only part of equation (1.10) required to determine corners. From this matrix the principal axes of the gradients are simply the eigenvectors of the matrix. If both eigenvalues of the matrix are large then the point is a corner. It is computationally difficult to solve for the eigenvalues, therefore; a simpler computation is to use a combination of the determinate and the trace of the matrix.

$$Tr(M) = \lambda_1 + \lambda_2 = A + B = \overline{I_x^2} + \overline{I_y^2} \tag{1.13}$$

$$Det(M) = \lambda_1\lambda_2 = AB - C^2 = \overline{I_x^2}\,\overline{I_y^2} - \overline{I_x I_y}^2 \tag{1.14}$$

Harris and Stephens then present the following formulation for a corner response term.

$$R = Det(M) - kTr(M)^2 \tag{1.15}$$

Harris and Stephens do not provide further information about this equation such as how it was derived or the reason this is a good corner response. Therefore, in the following section this equation will be further explored.

Harris and Stephens developed this response function to be able to detect both corners and edges at the same time. Using parameter k, the sensitivity to corners or edges can be changed. Increasing the value of k increases the sensitivity to edges and decreases the sensitivity to corners. Decreasing k increases the sensitivity to corners and decreases the sensitivity to edges. If this

detector is used simply to find corner points empirical testing in other research has determined that values between 0.04 and 0.08 produced good results.

1.2.2 Laplacian of Gaussian

Using the Laplacian of Gaussian as an image filter was introduced in research concerning human vision [10]. This research also introduced the idea of using the Difference of Gaussians to approximate the Laplacian of Gaussian function. Marr [7] applied the research of the human vision to develop methods of edge detection based on the Laplacian of Gaussian convolution kernel. The two dimensional extension produces spots where the standard deviation of the Gaussian defines the size of the detected spots. Lindeburg [3] uses the Laplacian of Gaussian (LoG) as a basis to detect blob-like structures. Because the LoG is not separable the Difference of Gaussians (DoG) [7] can be used to approximate the LoG. This can be implemented by two separable Gaussian kernels; resolving the issue of separability of the LoG kernel. Lowe [11] uses this approximation method to implement the SIFT detector based on this scale-space technique.

Directly using the Laplacian of Gaussian kernel is computationally expensive, therefore; the Difference of Gaussian has become the preferred method of implementation.

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1.16}$$

Equation (1.16) is the non-separable Laplacian of Gaussian equation which can be approximated by the difference of two Gaussian functions.

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{1.17}$$

Equation (1.17) is the function for a two dimensional Gaussian which is well known to be a separable kernel. Using a separable convolution kernel means that it can be reduced to two one dimensional convolution kernels which can be added together to produce the two dimensional result.

$$DoG \triangleq G_{\sigma 1} - G_{\sigma 2} = \frac{1}{\sqrt{2\pi}}\left[\frac{1}{\sigma_1}e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{\sigma_2}e^{-\frac{x^2+y^2}{2\sigma_2^2}}\right] \tag{1.18}$$

Equation (1.18) is the Difference of Gaussian convolution kernel which is the combination of two Gaussian kernels. This method is often used to improve the computational performance of the filter.

Principles of human vision can be directly linked to the Laplacian of Gaussian function for detection of image structure [10]. This leads to natural methods for analyzing images. Lindeburg develops this technique to locate blob-like structures to determine an appropriate scale to analyze an image. Lowe uses the scale information to locate robust feature points in an image. Each

method uses the Difference of Gaussian approximation which produces equivalent results to locate blob-like structures in an image.

1.2.3 Control Point Detection Performance

Image registration depends on control points in two images that represent the same point in the scene. This analysis is focused on the effect of noise on the Harris and LoG detectors. Number of corresponding points, Localization error, and Repeatability are used as performance measures to characterize the point detectors. Number of corresponding points is the number of points in one image that are within 2 pixels of a point in another image. The localization error is the average distance between corresponding points. Repeatability is the number of corresponding points divided by the total number of points in the original image.

As the parameter k in the Harris corner detector is changed these performance measures are calculated. This process is repeated for the parameter σ in the LoG detector. These measures are calculated at successively larger noise levels to determine the effect of noise on the detector. The performance measures are used to develop conclusions about the effect of noise on the detector's performance.

1.3 Registration

In this study three registration methods are compared. First, projective registration, second, piecewise registration and finally weighted linear registration, each method of registration builds upon the previous method to improve the results. Different registration methods have different limitations and levels of complexity. Therefore, it is important to understand the method used to determine the projective registration which is the basis for piecewise and weighed linear registration.

Projective registration is based on linear geometric transformations of a plane which can be described in a concise mathematical form [1]. This technique considers each image as a single plane being viewed from different positions in space. Piecewise registration is a technique which is used to account for geometric distortions in the image [12]. These distortions are from objects which do not lie on a single plane such as a box which has one plane for each face. This method cuts the image into small triangles which can be registered using a linear affine transform creating a localized mapping that does not cover the whole image [13]. Weighted linear registration works to smooth the piecewise linear mapping to create a smooth global mapping and register the whole image domain.

1.3.1 Projective Registration

Projective registration maps different views of the same scene using the model of a plane. Using four points from each view, the mapping between the views can be calculated. This maps each point in one view to a point in the other view. This model of the scene is significantly limited because the scene is not a plane. It is a view of a three dimensional world with objects that are at many different planes. A projective model is only capable of registering to one plane, such as the background or the foreground of the scene. Piecewise and weighted linear registration methods are designed to improve upon this technique to resolve this issue.

1.3.2 Piecewise Linear Image Registration

Piecewise linear image registration is a technique by which mapping functions are developed for pieces of the image rather than a global mapping function [12]. A typical image registration algorithm will apply a single mapping function to the whole image domain. Localized distortions in an image are not able to be captured by global mapping functions. A projective transformation is an accurate global transformation when the geometric differences are negligible. However, 3-D structures in the scene will produce local geometric differences between images. Goshtasby developed methods for creating piecewise linear [12] and piecewise cubic [14] mapping functions when registering images with local geometric differences. The results demonstrate the capability of these methods to solve nonrigid image registration problems.

1.3.3 Weighted Linear Image Registration

Piecewise registration produces a mapping between the reference image and the sensed image by dividing the images into sections which can be independently registered. As a result only the area under the pieces will have mappings between the two images. Any area which remains outside the convex hull of the points will not be registered. Registering each piece independently produces sharp transitions between each mapping function. Weighted linear registration smoothly merges the mapping functions created by the piecewise registration method. It also extends the mapping functions to the area which is not registered by the piecewise method.

Chapter 2

Analysis of Harris and LoG Point Detectors

Locating points in an image is important in many computer vision applications. In image registration, these points are called control points, which are used to find the mapping function that brings the images into alignment. Corner points describe unique locations in an image where two gradients merge, such as the corner of a box. Spots in an image describe areas where objects of a certain size are present, such as a rock. Each application may require a different interest point detector to locate stable points that are high in information content. For example a man-made scene with edges maybe more suitable for a method which finds locations where the edges intersect. Points detected from this method can be classified as corners. While a natural scene containing small rocks or smooth objects may perform better with a blob detector.

2.1 Harris Point Detection

The Harris corner point detector evolved from the Moravec [8] detector to overcome some of its deficiencies. A significant problem with the Moravec detector is that it is non-isotropic. This issue arises from the fact that it only

evaluates variations every 45 degrees. Harris and Stephens developed the detector to simultaneously locate corner and edge points in an image in a single pass and some post processing to create edges from edge points. Based on this, they developed a simple yet powerful detector, which overcomes the weaknesses of the Moravec detector.

Some of the parameters in the corner detector derived by Harris and Stephens are not fully explained. A discussion is presented as to how the Moravec detector can be improved by using auto-correlation and, a matrix is developed to represent the curvature of the local gradient. The eigenvectors of the matrix represent the principal components of the gradient. In the final result the parameter k appears with little indication of where it originated. If an understanding of this term can be made, it will become useful in determining what value to use for k to obtain the best corners. This discussion will develop a derivation of the corner detector based on the concepts developed in the paper to demonstrate where the parameter k originates.

Comparing the eigenvalues of the matrix in equation (1.12) four domains can be defined. A region where two eigenvalues are small indicates that the surface is relatively flat and uniform; there are neither edges nor corners. Two edge regions indicating there is a strong gradient in only one direction is defined when one eigenvalue is large and the other is small. And a corner region when both eigenvalues are simultaneously large, indicating a strong gradient in two directions orthogonal to each other.

Solving for the eigenvalues is a computationally expensive problem which would be best to avoid if possible. Therefore, any solution should be derived from the trace and determinant since they are directly related to the eigenvalues but do not require explicitly deriving the eigenvalues. Using this idea there must be some method to derive the corner response function defined by Harris and Stephens. The derivation begins by evaluating the solution for the eigenvalues of the matrix M from equation (1.12).

The eigenvalues of a matrix are obtained from:

$$Det(M - \lambda I) = 0 \tag{2.1}$$

λ is a solution of this equation.

$$Det\left(\begin{bmatrix} A & C \\ C & B \end{bmatrix} - \lambda I\right) = 0 \tag{2.2}$$

Expanding equation (2.2) gives the quadratic equation:

$$(A - \lambda)(B - \lambda) - C^2 = 0 \tag{2.3}$$

$$AB - \lambda A - \lambda B + \lambda^2 - C^2 = 0 \tag{2.4}$$

This can then be rewritten in standard form

$$\lambda^2 - \lambda(A + B) + (AB - C^2) = 0 \tag{2.5}$$

The trace and determinant are defined as

$$Det(M) = AB - C^2 \tag{2.6}$$

$$Tr(M) = A + B \tag{2.7}$$

and can substituted into the equation (2.5) to give the following result

$$\lambda^2 - \lambda * Tr(M) + Det(M) = 0 \tag{2.8}$$

This equation can then be solved to obtain

$$\lambda_1, \lambda_2 = \frac{-Tr(M) \pm \sqrt{\big(Tr(M)\big)^2 - 4(1)\big(Det(M)\big)}}{2(1)} \tag{2.9}$$

Properties of the roots λ_1, λ_2 can be described using the discriminant of the quadratic equation.

$$D = \big(Tr(M)\big)^2 - 4(1)\big(Det(M)\big) \tag{2.10}$$

Since the discriminate describes the relationship between the roots it is particularly useful in this case. When D is large this indicates that at least one of the eigenvalues is also large, if D is near 0 this indicates that the eigenvalues are nearly equal. When D becomes negative this indicates the determinate dominates the solution. A larger determinate will indicate that both eigenvalues are simultaneously large because the determinate is the product of both eigenvalues.

Corners are strongest when both the eigenvalues are nearly equal, therefore, corner values exist when D is less than or equal to zero. Looking at

the definition of the discriminate of the quadratic formula, shows that the term a will produce larger negative values for D.

$$D = b^2 - 4ac \tag{2.11}$$

Equation (2.10) is rewritten to include this term the result is.

$$D = \big(Tr(M)\big)^2 - 4(a)\big(Det(M)\big) \tag{2.12}$$

Letting a go to infinity maximizes the weight of the determinate causing corner strengths to dominate the result.

$$\lim_{a\to\infty}\big[(Tr(M)^2) - 4(a)\big(Det(M)\big)\big] = -\infty \tag{2.13}$$

This gives a the range $[1, \infty)$, which is too large to use in a practical application. Another form of this equation is needed where the constant term as a more reasonable range. Dividing both sides of the equation (2.12) by $-4(a)$ gives the following.

$$\frac{D}{-4(a)} = Det(M) - \frac{1}{4(a)}\big(Tr(M)\big)^2 \tag{2.14}$$

The following substitutions can be made to combine terms.

$$R = \frac{D}{-4(a)} \tag{2.15}$$

$$k = \frac{1}{4(a)} \tag{2.16}$$

$$R = Det(M) - k\big(Tr(M)\big)^2 \tag{2.17}$$

Using equation (2.16) the range for k can be calculated based on the range of a which has been absorbed by k. Plugging in 1 and ∞ gives k the range $[0, \frac{1}{4}]$. This form of the equation produces positive values for corners and negative values for edges.

2.1.1 Performance Analysis

Repeatability, localization error, and the number of corresponding points were chosen as the measures to use to characterize the point detectors [9]. This analysis considers how the point detections are being affected by adding noise to the image. Corner points are located in the original image and compared to the points detected in the noisy images. Localization error, number of corresponding points and repeatability measures are generated for each case.

2.1.2 Algorithm

1. Find the corners in the original image.
    a. Apply the Harris corner detector and obtain the corner response map
    b. Find local maximum values in a 3 x 3 neighborhood

2. Add noise to the image.
3. Find the corners in the noisy image.
4. Find the points which are within 2 pixels of the corner points in the original image.
5. Calculate the repeatability, localization error and number of corresponding points.
6. Increase noise and go to step 2. If the maximum noise is reached, then increase k and go to step 1. If maximum k is reached, then terminate.

The first step in the algorithm is to find points in the original image to create the basis set. Comparing the detected corners to the corners detected in the original image determines how noise affects the ability of the Harris detector to locate the corners in a noisy image. Metrics are used to measure how noise affects the performance of the Harris corner detector. Repeatability is defined as the number of corresponding points obtained between the original and noisy image divided by the number of points detected in the original image. This is a measure that shows how well the detector is able to find the same point under different noise levels. Localization error is defined to be the average distance from the original points to the corresponding points detected in the noisy image. This shows how well the point detector is able to locate the correct position of a corner in the presence of noise. Number of corresponding points is defined as the total number of points which are located within 2 pixels of the original points. This gives a good measure of the ability of the detector to find corner points in the noisy image.

2.1.3 Results

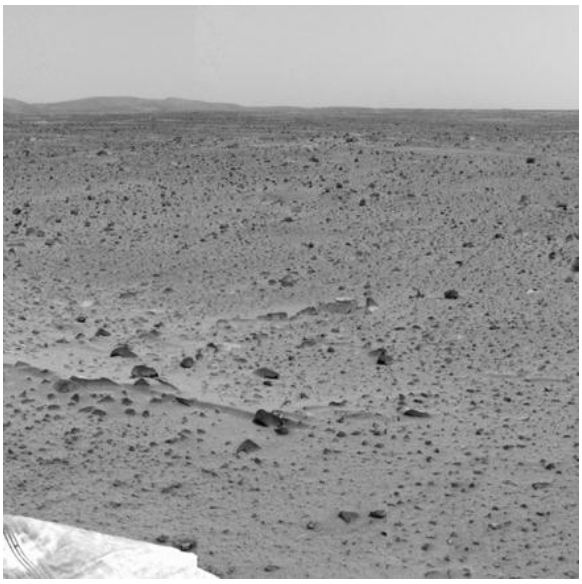

Figure 2.1 - Mars1.pgm

Figure 2.2 - Pentagon.pgm

The Mars image (Figure 2.1) and the Pentagon image (Figure 2.2) are the original images used in this experiment.

2.1.4 Data Analysis

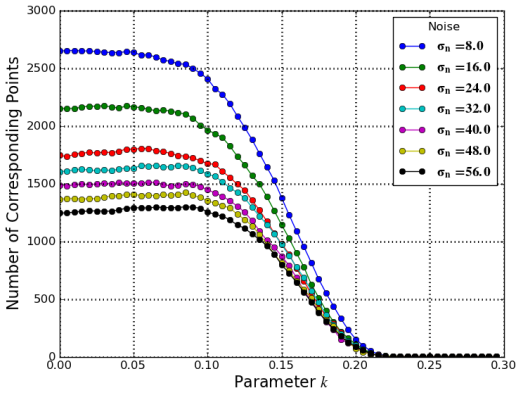

Figure 2.3 : Mars Number of Corresponding Points

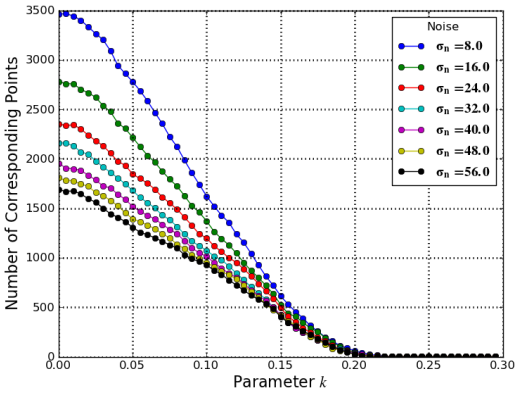

Figure 2.4 : Pentagon Number of Corresponding Points

Figure 2.3 and Figure 2.4 show that the number of corresponding points is affected significantly by the value of k. As expected, increasing k reduced the number of points as the response of the detector becomes more negative. This also shows that the Mars image produces corners which are similar in strength because the number of points is relatively flat until about $k = 0.1$. Using $k = 0$ gives a near maximal number of points in both images. The Pentagon image shows most of the points being removed at much smaller k values. Because k shifts the response from positive to negative values, this indicates that the majority of points most likely lie along edges.

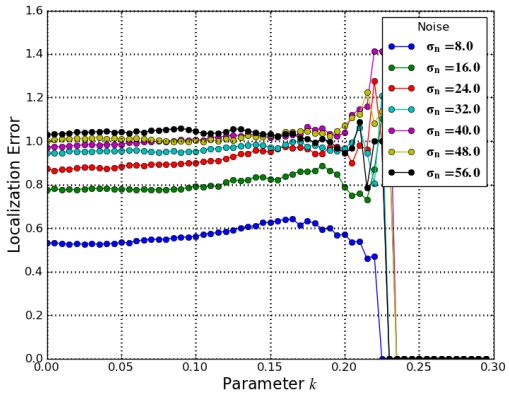

Figure 2.5 : Mars Localization Error

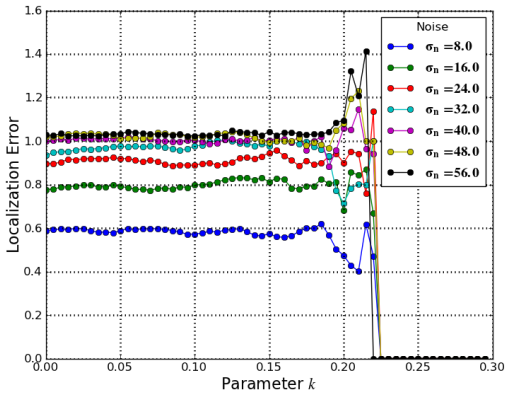

Figure 2.6 : Pentagon Localization Error

Localization error gives a better view about the quality of the detections regardless of the corner response value or the number of points. This shows how close the point detections are to the true point locations. Increasing the k value did not move the points significantly until it approached the limit of 0.25 at which time all the points are filtered out.

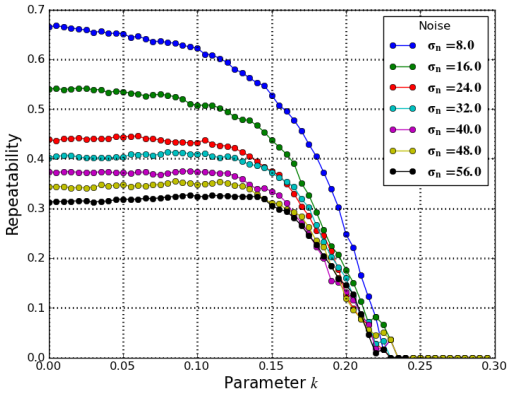

Figure 2.7 : Mars Repeatability

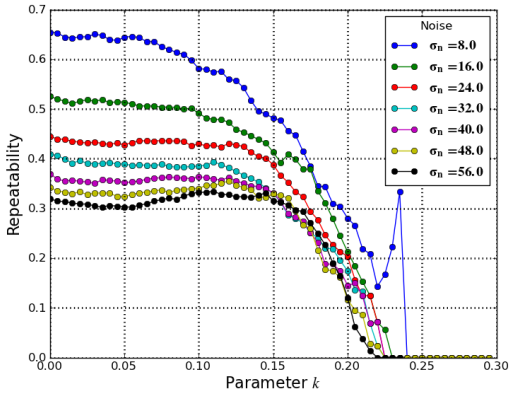

Figure 2.8 : Pentagon Repeatability

Repeatability gives the percentage of points that are found in the image. Increasing k generally decreases the repeatability for both the mars and pentagon images. This indicates that as response is being shifted towards edges the results are becoming less stable for corner detections. When k approaches 0.25 the number of points found become very small, ten or fewer points, this causes the results to become unstable, showing an increase in repeatability.

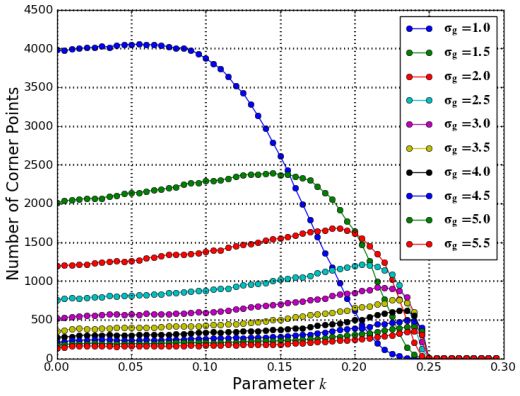

Figure 2.9 : Mars Number of Corner Points

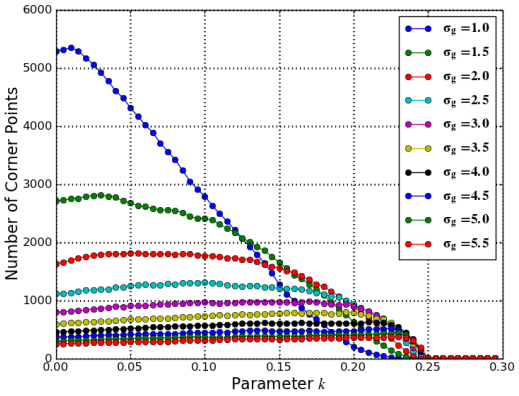

Figure 2.10 : Pentagon Number of Corner Points

Detecting corners only makes sense when looking at a neighborhood of pixels. Increasing the size of the neighborhood affects the size of the corners being detected. In this case the neighborhood of pixels considered is generated from the Gaussian smoothing which is performed on the x and y gradients. Increasing the σ term in the Gaussian smoothing increases the size of the neighborhood used to determine corners. Figure 2.9 and Figure 2.10 show the number of corner points found in the images as σ is changed.

2.2 Laplacian of Gaussian Point Detection

The Laplcain of Gaussian is a much simpler detector which can be used to find spots or blobs in an image. This detector is the basic blob detector where the standard deviation of the Gaussian specifies the size of the blobs to detect. The detector at varying standard deviations can detect objects at different scales in the scene.

2.2.1 Algorithm

1. Find the spots in the image for the given σ
    a. Apply the LoG and obtain the response image
    b. Find locally maximum values using 3 x 3 neighborhoods.
2. Add noise to the image.
3. Find the spots in the noisy image.
4. Find points that are within 2 pixels of the points in the original image.
5. Calculate the repeatability, localization error, and number of corresponding points.
6. Increase the noise and go to step 2. If the maximum noise is reached go to step 1. Stop when the maximum σ is reached.

2.2.2 Results

Spot detection has been applied to many applications and has been found to produce satisfactory results. It produces points that are stable with respect to rotation at a given scale. Further scale-space analysis is used to make it stable with respect to scale [6]. The following analysis shows other properties of the Laplacian of Gaussian detector.

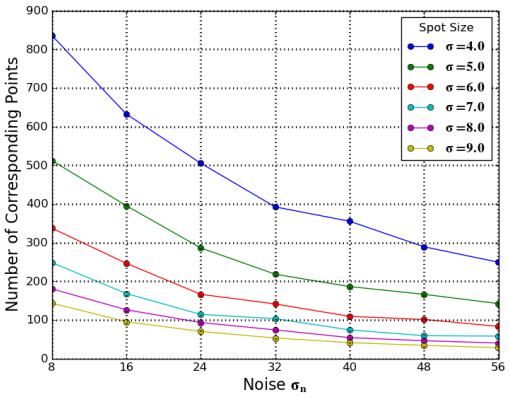

Figure 2.11 : Mars LoG Number of Corresponding Points

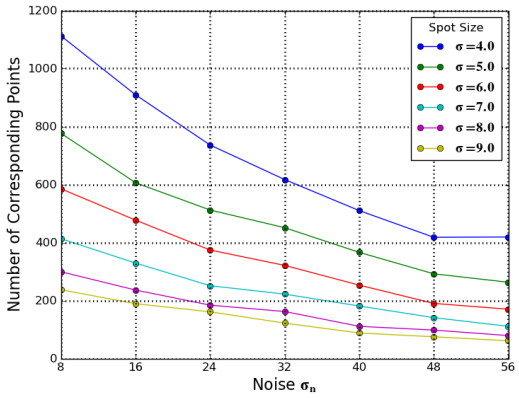

Figure 2.12 : Pentagon LoG Number of Corresponding Points

Increasing noise decreases the number of corresponding points. Using a larger scale for the spot size will also decrease the number of corresponding points. As it would be expected, the LoG detector is affected by noise just as the Harris detector.

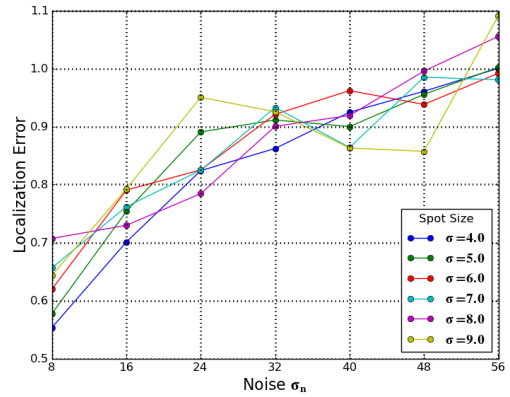

Figure 2.13 : Mars LoG Localization Error

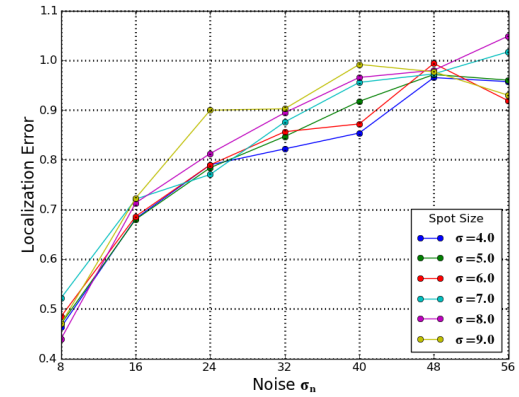

Figure 2.14 : Pentagon LoG Localization Error

Localization error shows how correct the point locations were. Increased localization error indicates that the detections are becoming less reliable. Results demonstrate that increasing the noise has a significant effect on the

ability of the detector to accurately locate points. Using a different spot size for the LoG detector also has an effect on the localization error at a specific level of noise. This is because objects of significance are at some particular spot size. Smaller blobs will be randomly joined to other blobs when the scale is increased. However, the general trend will be that localization error will increase as noise increases.

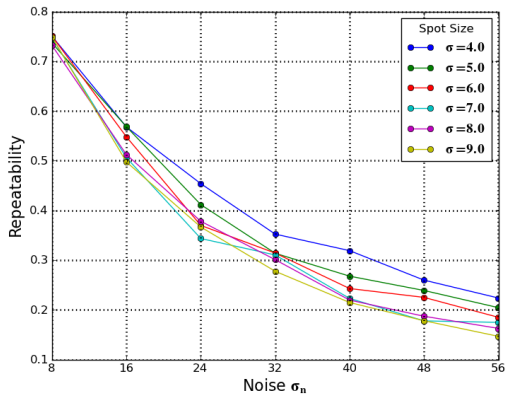

Figure 2.15 : Mars LoG Repeatability

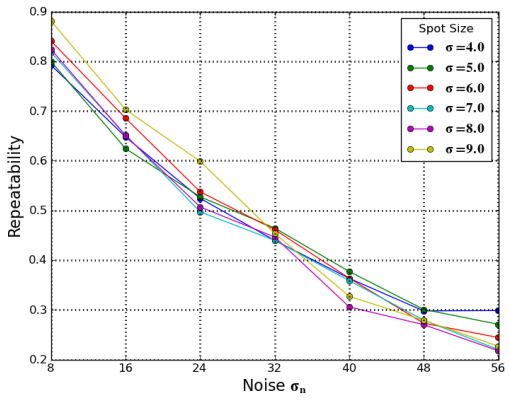

Figure 2.16 : Pentagon LoG Repeatability

Repeatability shows that noise dramatically affects the performance of the LoG detector when there is noise in the image. Increasing noise quickly decreases the repeatability which demonstrates that the detector is not able to locate corresponding points in images in the presence of noise. These results show that the LoG and the Harris detectors are both significantly affected by noise. Neither detector is inherently better at locating points when noise is present in an image.

Chapter 3
Projective Image Registration

Image registration is the process of spatially aligning two or more images. Spatial alignment is achieved using 2D geometry transforms, such as rigid, similarity, affine, and projective. A rigid transform has three degrees of freedom: rotation and translation along each axis. The similarity transform extends the rigid transform by scale, giving the transform four degrees of freedom. An affine transform adds non-isotropic scaling and shearing, adding two more degrees of freedom and giving this transform six degrees of freedom. The projective transformation adds the ability to model vanishing points and adds two more degrees of freedom, giving this transform eight degrees of freedom [1]. These transformations can be written as a 3 x 3 matrix, commonly called a homography.

Point transformations can be written as,

$$x' = Hx, \tag{3.1}$$

where x is a homogenous point in the reference image and x' is a point in the sensed image. The homography gives the relationship between the two points. The homography H is given by:

$$H = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}. \tag{3.2}$$

There are only eight unknowns to find.

Point x is a control point in the reference image and, x' is a control point in the sensed image. These are corresponding control points representing the same scene point as seen from each view. The set of corresponding control points can be written as [13],

$$\{(x_i, y_i), (X_i, Y_i) : i = 1, \dots, N\} \tag{3.3}$$

(x_i, y_i) is the i^{th} control point in the reference image and (X_i, Y_i) is the i^{th} control point in the sensed image. Mapping functions can be written to represent the relationship between points in the reference image and the corresponding control points in the sensed image.

$$f_x(x_i, y_i) = X_i \tag{3.4}$$

$$f_y(x_i, y_i) = Y_i \tag{3.5}$$

These mapping functions define two single-valued surfaces interpolating 3D points,

$$\{(x_i, y_i, X_i) : i = 0, \dots, N\} \tag{3.6}$$

$$\{(x_i, y_i, Y_i) : i = 0, \dots, N\}. \tag{3.7}$$

Using these control points the images can be registered by determining the mapping functions that represent the surfaces best fitting these control points. Knowing the corresponding points, the task of finding the solution for the homography becomes the task of solving a set of simultaneous equations. This can be done by solving the linear equation $Ax = b$. Finding the solution to a system of equations is not difficult, however; finding correct corresponding control points is.

To automatically locate corresponding control points in two images, a set of control points should be found in each image first. From the control points the set of corresponding control points can then be determined. There exist many methods to solve this problem. In this study template matching is used as the basis for solving this problem. With template matching a best guess for each correspondence can be found. From this best guess of corresponding control points, errors are then removed. A random sample and consensus (RANSAC) method is used at this point to remove points which do not fit the dataset. The resulting set of correspondences is considered to be the set of actual corresponding control points.

3.1 Control Points

Locating control points in each image is the first step in finding the correspondence between points in the images. These points identify locations in

the images where there is some unique structure. In this study the Harris detector is used to find corner points.

Figure 3.1 : pentagon1.pgm

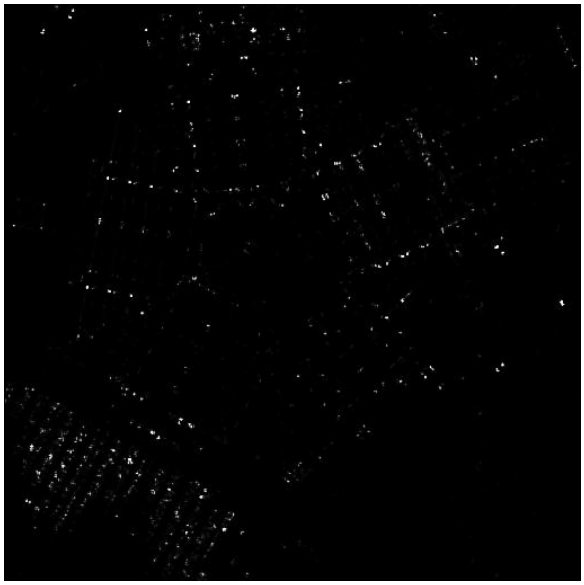

Figure 3.2 : corner map for pentagon1.pgm

Figure 3.2 shows the corner map generated by applying the Harris corner detector to the Pentagon image shown in Figure 3.1. Locally maximum corner response values are taken as points and added to a list of control points. This can easily result in a large number of points. However, only the strongest points are needed. Sorting the list in descending order and taking the set of points from the beginning of the list gives a set of the strongest control points.

3.2 Template Matching

Template matching is a powerful technique to find the location where a small template best matches a larger image [15]. This requires that the scene in

both images be of the same scale and orientation. Good matches can still be obtained with slight variations. However variations of the scale and rotation will likely cause the matching to be inaccurate. Scene lighting can also have a significant effect on the matching process. However, using the gradient of the image for the template matching process can significantly reduce the effects of the scene lighting; edges of shadows may still be a problem.

Template matching is a simple technique of comparing the template centered at each pixel in the image. Depending on the metric used to calculate the match, the best match will either be the largest or smallest result of the function. Using cross correlation, the match rating is computed from,

$$R(x,y) = \sum_{x',y'} \left(T(x',y') * I(x+x', y+y') \right), \tag{3.8}$$

$R(x,y)$ is the template response at a given location in sensed image I when the template T from reference image is centered at the pixel in image I. This produces a response image where the global maximum value in the image is the location where the template best matches the image. Template size needs to be sufficiently large to produce a unique match. For the type of images used templates of size 15 x 15 pixels are found to perform well. Each template can then be created by taking a 15 x 15 pixel subimage centered at each interest point.

Figure 3.3 : Pentagon 1

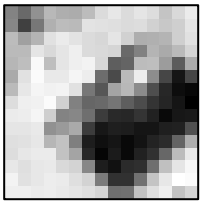

Figure 3.4 : 15 x 15 pixel template from Pentagon 1

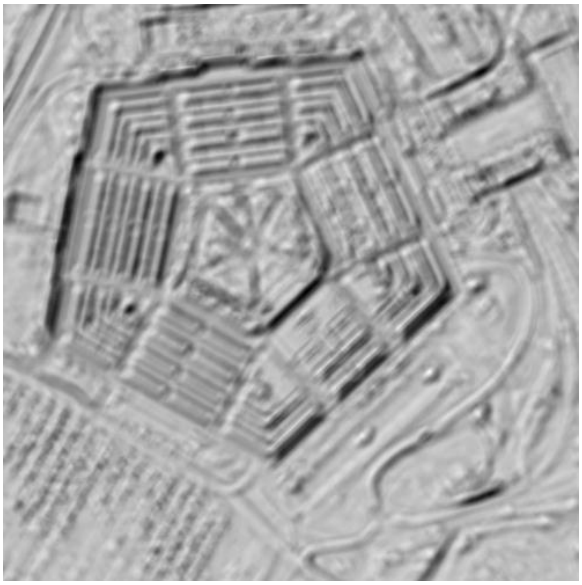

Figure 3.5 : Template response on Pentagon 2

Figure 3.6 : Pentagon 2

Figure 3.3 is the image from which a template is extracted. The boxed area in the image is the template that is searched in the image in Figure 3.6. Figure 3.4 is an enlargement of the 15 x 15 pixel template taken from Figure 3.3.

This template is evaluated at each pixel in the image to create the response image shown in Figure 3.5. The location of the globally maximum value in the response image is the location where the template best matched. Figure 3.6 shows the location of the best match for the template which is highlighted by the box representing the area of the best match for the template.

3.3 RANSAC Matching

Random sample and consensus (RANSAC) was introduced by Fischler and Bolles [4] in 1981 and has become a fundamental component of many matching or fitting algorithms. Algorithms which operate on sampled data often need to be able to fit the data to some model such as a line or curve. Sampled data often contains data points that represent gross errors that need to be removed from the data set to correctly model the real data. This is a filtering step that can be applied to data where the data is expected to fit to some model such as a line or a higher order curve. After the gross errors are removed from the data set, more traditional methods can be applied to the data to fit a line or curve to the data by using methods such as least squares. RANSAC is a general method that can be used to find corresponding points that match with the projective transformation model [16].

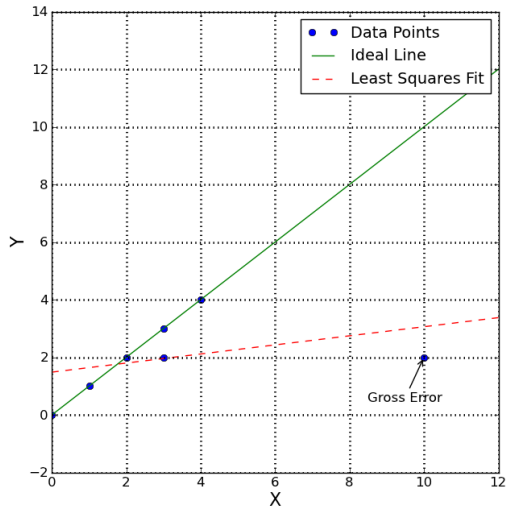

Figure 3.7 : Data set with gross error

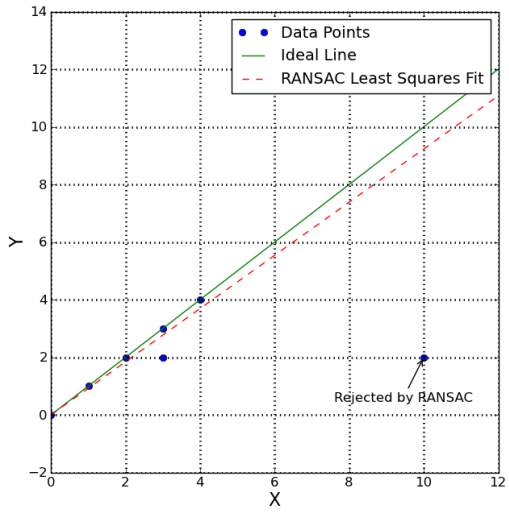

Figure 3.8 : Data set with error removed

Figure 3.7 demonstrates the results of fitting a line using least squares to a data set with a gross error. This error significantly skews the results of the least squares approximation away from the ideal solution. In this case applying RANSAC to find and remove this error will significantly improve the result. Using the RANSAC method is straightforward. First, randomly select two points which can be used to construct a line. Second, find the distance between each point and the line; points which are within a threshold distance to the line are in the consensus set. If there are enough points in the consensus set then use those points to find the final solution. If there aren't enough points in the consensus set then a new randomly selected set of points is tried. Figure 3.8 shows the result of using RANSAC on the data. In this case the point at $(10,2)$ is rejected because it is outside of the acceptable error allowed to fit the line model. Using the RANSAC method is powerful, as it is able to remove those data points that

contain a significant positional error and should not be included in the data, which is averaged to approximate the solution.

To apply this method to projective registration, the model used in data fitting is the projective transform function. Four points are randomly selected and used to calculate the parameters of a possible projective homography. This homography is applied to all of the remaining points. Each of the points is tested to determine the ones that lay within some error tolerance of their corresponding points. When a homography is found that satisfies a large enough set of corresponding points, the remaining points are determined to be erroneous points and are removed from the set of corresponding points. Least squares can then be used on the remaining points to find the best transformation parameters that fit the points.

3.4 Projective Transformation

A projective transform models the view of a plane in 3D from different points of view. The homography is used to model the relationship between two different views of the plane. Points from one view have corresponding points in the other view. From these correspondences the projective homography parameters can be determined. Once the homography is known, every point on the plane in one view can be transformed to match another view of the plane. Using equation (3.1) H can be solved using four known corresponding points.

A system of equations for the corresponding points can be written as follows [16],

$$X_i = ax_i + by_i + c + 0 + 0 + 0 - gx_iX_i - hy_iX_i \tag{3.9}$$

$$Y_i = 0 + 0 + 0 + dx_i + ey_i + f - gx_iY_i - hy_iY_i. \tag{3.10}$$

Each point gives two equations, one for each coordinate. Writing the system of equations in matrix form gives the following [16],

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -y_1X_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1Y_1 & -y_1Y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -y_2X_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2Y_2 & -y_2Y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -y_3X_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3Y_3 & -y_3Y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -y_4X_4 \\ 0 & 0 & 0 & x & y & 1 & -x_4Y_4 & -y_4Y_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \\ X_4 \\ Y_4 \end{bmatrix}, \tag{3.11}$$

which can be solved to determine the parameters of the projective transformation.

Once the parameters have been determined, the sensed image is resampled and overlaid onto the reference image. This is done by finding the location of each pixel in the reference image in the sensed image. The intensity in the sensed image is then read and saved at the pixel in the reference image. In this manner, the sensed image is resampled point by point to align with the reference image.

The components of the projective transformation that relate the coordinates of points in the reference image to those of the sensed image are:

$$f_x(x,y) = \frac{ax + by + c}{gx + hy + 1} \tag{3.12}$$

$$f_y(x,y) = \frac{dx + ey + f}{gx + hy + 1}. \tag{3.13}$$

Equation (3.12) is used to find from a point in the reference image the X coordinate of the same point in the sensed image. Equation (3.13) is used to find from a point in the reference image the Y coordinate of the same point in the sensed image. Using these two equations each point the reference image can be transformed to a point in the sensed image. The mapping functions in image form are shown in Figure 3.9 and Figure 3.10 in twenty five discrete levels.

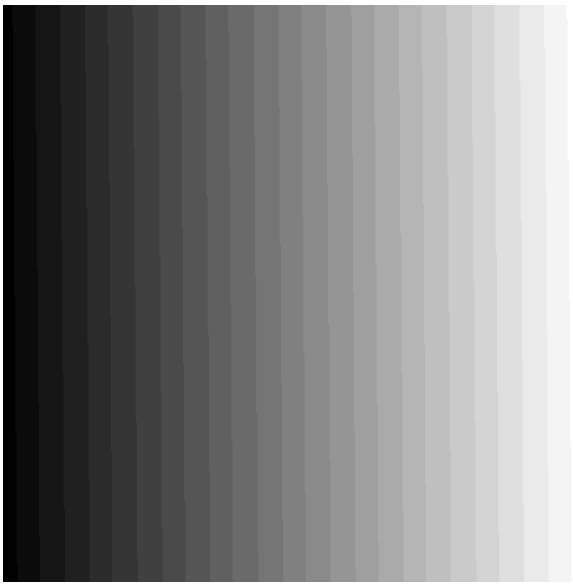

Figure 3.9 : $\boldsymbol{f_x(x,y)}$

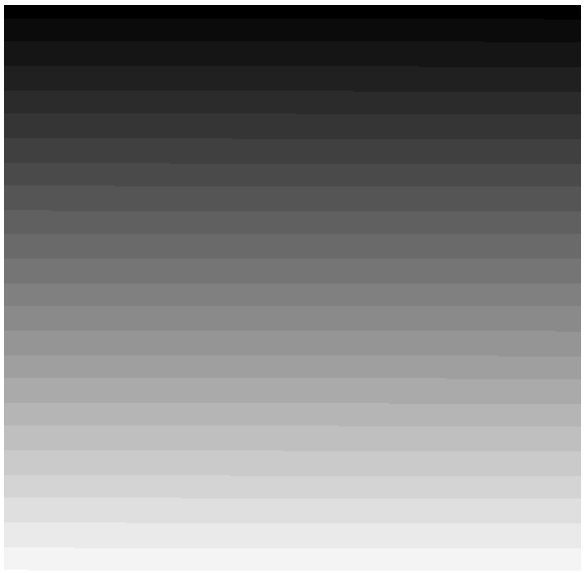

Figure 3.10 : $\boldsymbol{f_y(x,y)}$

3.5 Projective Registration Results

Figure 3.11 : Registered Pentagon images

Figure 3.12 : Difference of the registered images

Figure 3.11 shows overlaying of Pentagon 2 image and Pentagon 1 image using the projective transform determined by the method outlined above. Figure 3.12 shows the absolute difference of registered images. The ground is well registered but the top of the building is not. Projective geometric difference between the images is large enough to cause the errors. To improve these results piecewise linear registration is used to compensate for local geometric differences between images.

Chapter 4
Piecewise Linear Registration

Projective registration is powerful and flexible; however, it only models changing views of a plane. It does not model real world scenes where objects are not located on a plane [12]. An image represents just one view of a three dimensional scene. In order to register different views of a 3D scene, projective registration needs to be extended to accommodate the geometric difference between different views. Piecewise linear registration approaches this problem by cutting the image into triangles each of which can be registered by an affine transformation [12].

Piecewise linear registration can be broken down into six steps: 1) scale reduction, 2) finding corresponding points in the reduced scale image, 3) finding corresponding points at a larger scale,4) triangulating the points in full scale, 5) calculating affine parameters for each triangle, and 6) registering each triangle. Scale reduction is used to effectively reduce the geometric distortion. Corresponding points are found between lowest scale images using a single projective model because the geometric distortions have been sufficiently reduced. These corresponding points then are located in each successively larger scale until they are located in the full scale image to allow them to drift

away from a single projective model to many affine models. Once the points are located in the original images the points are triangulated. Each corresponding triangle requires an affine registration. Finally, registered triangles are pieced together to create the final registration.

4.1 Steps for piecewise linear registration

1. Reduce the image scale sufficiently to considerably reduce their geometric differences
2. Find corresponding points in the images
3. Track corresponding points in successively high scales
4. Triangulate the points in full scale images
5. Calculate the affine parameters for each corresponding triangle
6. Register corresponding triangles

4.2 Scale Reduction

Image pyramids [17] are typically used in scale space analysis [3], and have been applied to feature point detection [6]. Each level in the pyramid reduces the scale by a factor of two from the previous level. The top of the pyramid is the lowest scale while the bottom of the pyramid is the original image. Reducing the image scale reduces the geometric difference between images. This allows a projective model to find corresponding points in the images.

4.3 Tracking corresponding points

Once the image pyramid is created and corresponding points are determined in the images at the top of the pyramid using the projective model. These points need to be tracked through each level in the pyramid until they are located in the original images at the bottom of the pyramid. Corresponding points need to track corresponding features, therefore; point locations cannot be simply scaled by a factor of two from one level to the next. This is due to the location of the features moving from level to level. Instead, the points are tracked from one scale to the next.

Correctly tracking a point through different scales poses uncertainty as points in higher scales no longer follow a single projective model. Points located in the images at the top of the pyramid follow the projective model; however, as the points move down the pyramid they diverge from the projective model. As points are moved from level to level, it is required to determine whether the points are in fact following the features that they are supposed to follow. If a point is not tracked correctly, the corresponding points in the images may represent different points in the scene. Template matching was used to verify the correctness of the correspondences.

Feature tracking errors can occur when the features begin tracked belong to a line. A line will become thicker when moving down the scale pyramid, causing the corner points at the end of the line to separate. Lines will also grow in length, therefore; points may appear to be at opposite ends of the line between

different views of a scene. Errors such as this need to be corrected or removed; otherwise, they will produce useless results. In this study the feature at each point is compared to the feature at the corresponding point in the other image to verify the correctness of a match. If the templates are not sufficiently similar, the correspondence is removed.

4.3.1 Algorithm for tracking the points

1. Multiply the point location by factor of two to find the initial position
2. Create a template from the source image at the new point location
3. Find the best template match in the second image within a 3 x 3 neighborhood of the corresponding point in the second image
4. Use this new position as the center of a template from the second image
5. Compare this new template back to the template from the source image
6. If these templates are not sufficiently similar, remove the points from the list of corresponding points

4.3.2 Tracking corresponding points with increasing scale

Figure 4.1 : Pentagon 1 image at 1/4 scale

Figure 4.2 : Pentagon 2 image at 1/4 scale

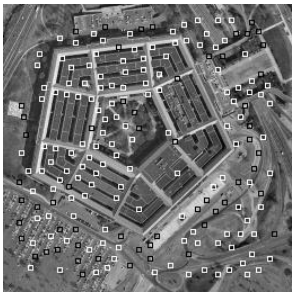

Figure 4.3 : Pentagon 1 image at 1/2 scale

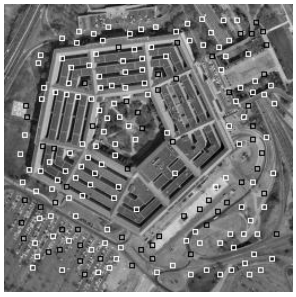

Figure 4.4 : Pentagon 2 image at 1/2 scale

Figure 4.5 : Pentagon 1 image at full scale

Figure 4.6 : Pentagon 2 image at full scale

Corner points detected in the image at the top of the pyramid are shown in Figure 4.1; this is the smallest scale used. Corresponding points for the image in

Figure 4.1 are shown in Figure 4.2. Point locations are determined at each lower level in the scale pyramid until they are determined in the original image at the bottom of the pyramid. Figure 4.5 and Figure 4.6 show the corresponding points located in the original images.

4.4 Triangulation of the points

Once corresponding points are determined in the original images the points need to be connected in a way to divide the image into small sections. Triangulation is the simplest geometric form which can be used to form regions that can be registered. Delaunay triangulation is often used to subdivide a domain into triangular regions because it avoids triangles with acute angles [18]. This method optimizes the triangles by maximizing the minimum angle of the triangles [18].

Given an arbitrary triangulation of a set of points where no triangle overlaps another triangle, the triangulation can be converted to Delaunay triangles. A triangle is Delaunay when the circumcircle of the triangle contains no other points [18]. When a point does exist in the circumcircle then the triangulation needs to be converted. Since the points have a valid triangulation, the point is a part of a triangle that shares an edge with the triangle being tested. Simply flipping the interior edge that is common in both triangles will create a Delaunay triangle [18]. This can be shown graphically in Figure 4.7. A triangulation that is not Delaunay and has a point within the circumcircle of the

triangle BCD is shown. Figure 4.8 shows flipping the interior edge to create a Delaunay triangle such that the circumcircle of ACD does not contain any other point.

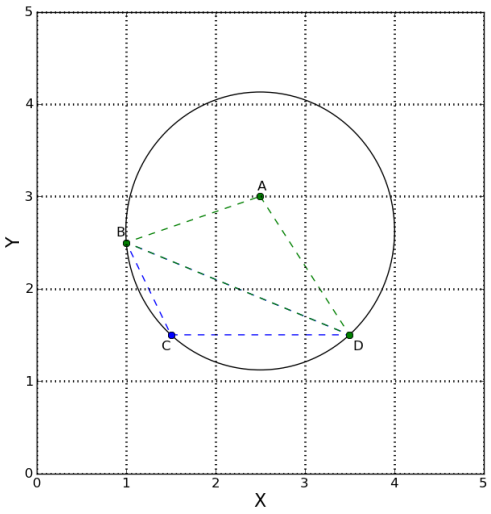

Figure 4.7 : Invalid Triangulation

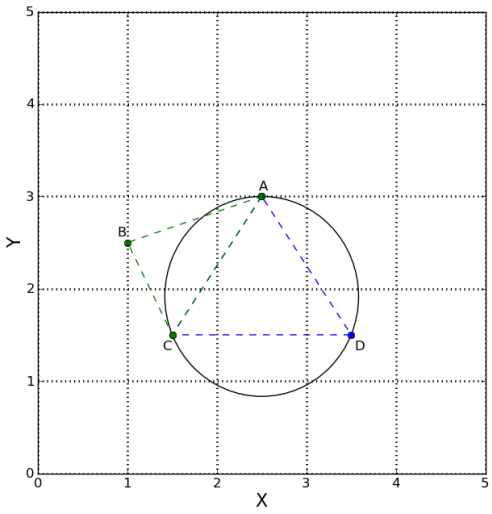

Figure 4.8 : Edge flipping to create valid triangulation

Figure 4.9 and Figure 4.10 show the results of the triangulation in the two images of the pentagon. It can be seen that there are convexities in the resulting triangulation which occur due to computational errors. These triangles have very small angles approaching zero, this causes these points to appear collinear and, therefore, is a degenerate case and cannot form a triangle. Triangles along the convex hull can become very flat; this happens when the center of the circumcircle is outside the convex hull where there are no more points to work with.

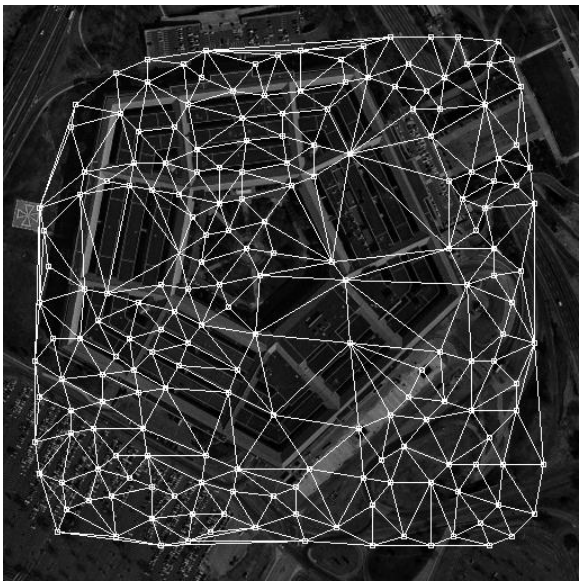

Figure 4.9 : Triangulation for Pentagon 1 image

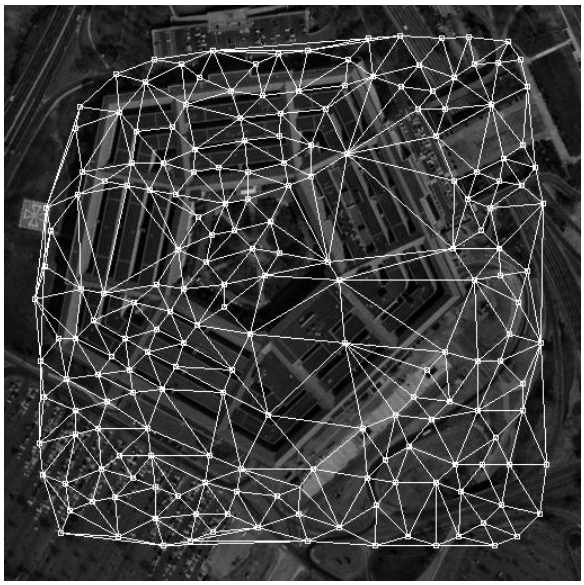

Figure 4.10 : Triangulation for Pentagon 2 image

4.5 Register Corresponding Triangles

Each triangle can be registered to its corresponding triangle using an affine transformation determined from the three vertices of corresponding triangles [13]. This generates a transformation function for each corresponding triangle. The final mapping function is created by piecing together each of the mapping functions from each of the triangles into a single mapping function. This is then a piecewise mapping function which accounts for geometric differences between the images.

An affine transformation is defined by [16],

$$H = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.14}$$

There are six unknowns which need to be determined. Because each corresponding point produces two equations, only three points are needed to solve for the unknown terms. The equations for the solution are given as [16],

$$X_i = ax_i + by_i + c + 0 + 0 + 0 \tag{3.15}$$

$$Y_i = 0 + 0 + 0 + dx_i + ey_i + f \tag{3.16}$$

This system of equations can be solved in matrix form,

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \end{bmatrix}. \tag{3.17}$$

Knowing the registration parameters $a, b, c, d, e,$ and f the following equations can be used to transform any point in a reference triangle to a point in the corresponding sensed triangle. For the i^{th} corresponding triangles, we have

$$X_i = a_i x + b_i y + c_i, \tag{3.18}$$

$$Y_i = d_i x + e_i y + f_i, \tag{3.19}$$

where i denotes the i^{th} triangle and is only valid for the area that is covered by the triangle. Using these equations each corresponding triangle can be

registered independently of the other triangles. These mapping functions for the Pentagon images are shown below in discrete form for viewing.

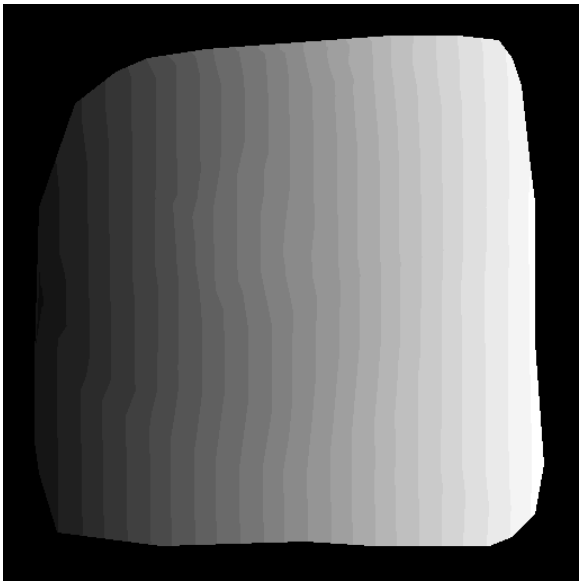

Figure 4.11 : $\boldsymbol{f_x(x,y)}$

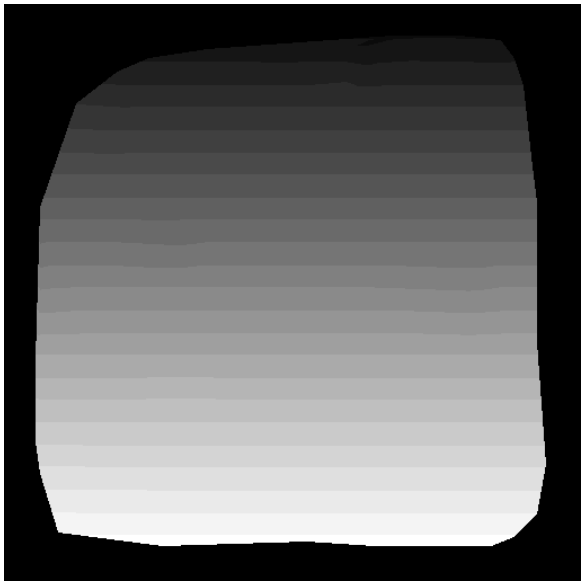

Figure 4.12 : $\boldsymbol{f_y(x,y)}$

Figure 4.11 and Figure 4.12 represent the components of the global transformation function after piecing together transformation functions of the individual triangles. The images are enhanced by showing only twenty five discrete levels in the final map and make the nonlinearity of the result more visible. Each triangle produces a separate affine mapping function, which is only valid within a triangle. Along the edges of each triangle the transformation map is not smooth because of the switching from one mapping function to another. This produces a transformation map that does not cover the whole image. Weighted linear registration discussed next extends this method to provide a smooth transition across triangle edges and extend the mapping function to the whole image domain.

4.6 Piecewise Linear Registration Results

Figure 4.13 : Piecewise linear registration: Image overlaying

Figure 4.14 : Piecewise linear registration: Absolute intensity differences

Figure 4.13 shows overlaying of the Pentagon images after piecewise linear registration. Figure 4.14 shows the absolute intensity differences of the overlaid the images. These results demonstrate registration of the top of the building as well as the ground surrounding the building and the court yard in the middle of the building. It can be seen that there are some errors in the registration where the images are not correctly aligned. Looking at the left side of the building it can be seen where the edge of the building from each image did not correctly align. In other parts along the edge of the inner buildings there are also areas where there is some error. These errors are, in part, due to the building being viewed from different views and occlusion is present along the boundary of the building.

Chapter 5
Weighted Linear Registration

Weighted linear registration extends the piecewise linear registration to improve the alignment process [13]. This method smoothly merges the triangles and extends the mapping function to cover the whole image domain [13]. Using the results of the piecewise registration method, each affine transformation is combined using a rational Gaussian basis function [13]. Each point in the reference image is transformed using a weighted sum of all the affine transforms created by the piecewise method. A Gaussian is used to create the weighting function so that only the neighboring triangles will affect the mapping function for a given triangle. This method will also extend the mapping function over the whole image domain.

5.1 Rational Gaussian Basis Functions

Each triangle contributes to the overall mapping functions by the following equations [13].

$$X(x,y) = \sum_{j=1}^{n} W_j(x,y) X_j(x,y) \tag{3.20}$$

$$Y(x,y) = \sum_{j=1}^{n} W_j(x,y) Y_j(x,y) \tag{3.21}$$

The mapping from a point in the reference image to a point in the sensed image is obtained from a weighted sum of all the affine mapping functions affecting that point. The affine mapping functions for the X and Y components of the j^{th} triangle are given as follows [13].

$$X_j(x,y) = A_j x + B_j y + C_j \tag{3.22}$$

$$Y_j(x,y) = D_j x + E_j y + F_j \tag{3.23}$$

Each weight function is defined by [13]:

$$W_j(x,y) = \frac{G_j(x,y)}{\sum_{k=1}^{n} G_k(x,y)} \tag{3.24}$$

The denominator is the sum of all the Gaussian functions at the point to normalize the sum of all the weight functions to one. Since the sum of all the weight functions is one the equations (3.20) and (3.21) become the weighted average of all the mapping functions when measured at a point. The Gaussian for the j^{th} triangle is defined by:

$$G_j(x,y) = \exp\left\{-\frac{\left(x-C_{x_j}\right)^2+\left(y-C_{y_j}\right)^2}{2\sigma_j^2}\right\}, \tag{3.25}$$

where

$$\sigma_j = sr_j. \tag{3.26}$$

Each triangle has a rational Gaussian weighting function that is centered at the centroid of the triangle, $\left(C_{x_j}, C_{y_j}\right)$. Equation (3.26) is the standard deviation of the Gaussian at the j^{th} triangle. Two terms are used, r_j and s to define the standard deviation of each triangle. The term s is a global parameter defining smoothness that determines the elasticity of the surface when following the piecewise linear map.

It has been suggested that r_j to be set to the radius of the circumcirle [13] of the j^{th} triangle however; this can cause a significant problem with the weighting functions. Since the circumcircle is a circle that passes through all vertices of a triangle, acute triangles will create large circles. Triangles which share an edge with the convex hull are often acute triangles and produce large circumcircles. In the following figure this is demonstrated using the triangulation created from the Pentagon 1 image.

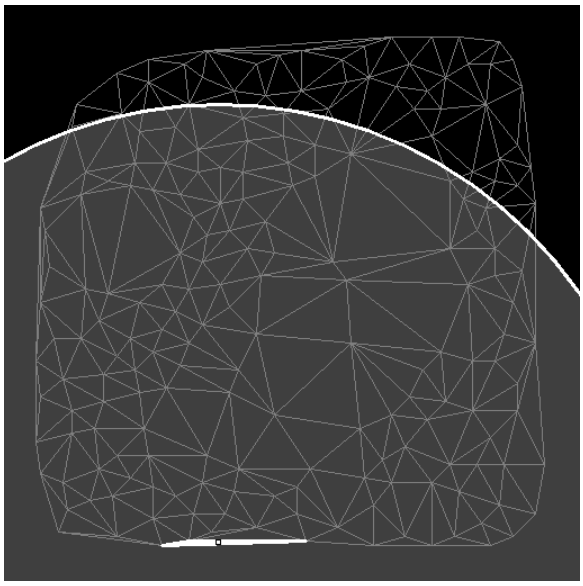

Figure 5.1 : Circumcircle of a thin triangle re-centered to the centroid of the triangle

Figure 5.1 shows the circumcircle which is created by the highlighted triangle and re-centered at the centroid of the triangle. If the radius of this circle is used for the r_j term of the standard deviation of the Gaussian function the Gaussian will have a large spread covering nearly the whole image. Using this would cause this one triangle to have a significant effect on triangles far beyond its neighbors causing these triangles to be pulled towards the transform generated from this triangle rather than the transform generated from their neighbors. To create a smooth surface each triangle should only affect its neighborhood.

One solution to this issue is to use the minimum distance between the centroid of the triangle and each vertex. Using this for the value of r_j gives thin triangles smaller values and larger triangles larger values. However, any solution which produces r_j values which only effect triangles nearby will be sufficient;

ideally only affecting neighboring triangles. In this study the r_j value is the minimum of the distance between the centroid and each vertex. Figure 5.2 shows the circle of radius r_j that is created using this method.

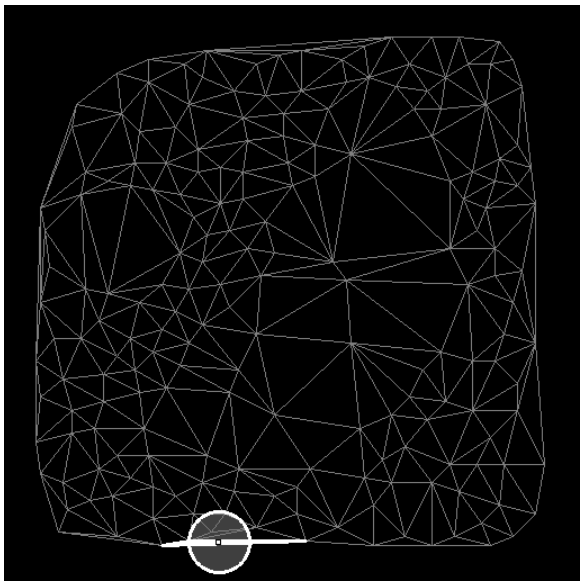

Figure 5.2 : Circle created from the minimum vertex distance centered at the centroid of the triangle

5.2 Weighted Linear Registration Results

Shown in the following figures is the result of applying the weighted linear registration method to the pentagon images. The image in Figure 5.3 shows the sensed image overlaid with the reference image. Figure 5.4 shows the absolute difference between the re-sampled sensed image and the reference image. To generate these results parameter s was set to 1.5. Affine mapping functions have been smoothly merged and extended over the whole image domain. Smoothing introduces errors when the edges of a triangle are coincident with the edges of a rigid object. In this scene the transition of the plane representing the

top of the building to the ground is discontinuous causing the smoothing errors to be visible.

Figure 5.3 : Weighted linear overlay result

Figure 5.4 : Weighted linear absolute differences

Mapping functions X and Y, equations (3.22) and (3.23), are shown in image form in Figure 5.5 and Figure 5.6. The mapping is warped to compensate for the geometric differences between the images. These images have been enhanced by only showing 25 discrete levels in the mapping function. This makes the curvature of the mapping more clearly defined for better visibility of the mapping function. This shows the smooth transitions between the affine mapping functions of adjacent triangles.

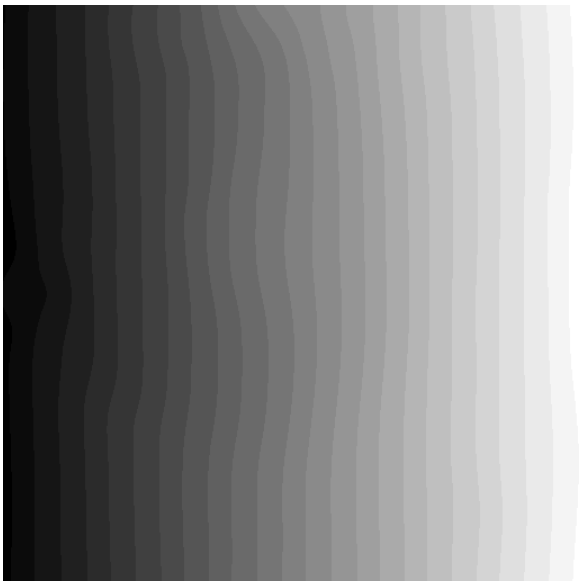

Figure 5.5 : $\boldsymbol{X}(\boldsymbol{x},\boldsymbol{y})$ map

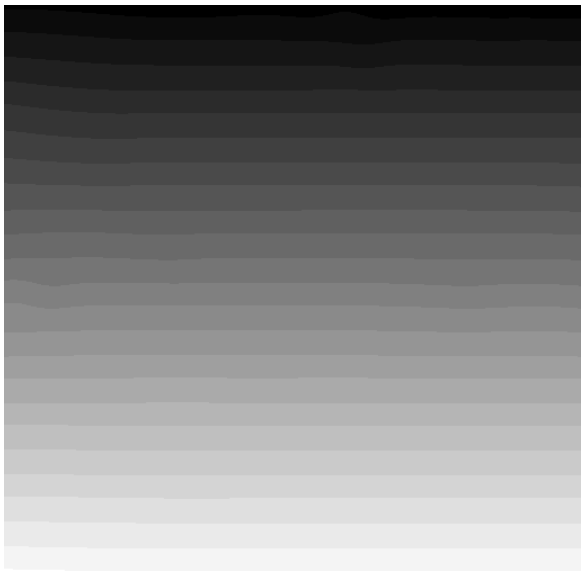

Figure 5.6 : $\boldsymbol{Y}(\boldsymbol{x},\boldsymbol{y})$ map

Chapter 6
Results

Each registration method was applied to a number of different data sets. These results demonstrate the improvements that can be archived by using the implemented nonrigid registration. It is also shown that this nonrigid method is sensitive to errors among corresponding points. In the piecewise linear method each mapping function is forced to pass through the control points. Whereas in projective registration the errors are averaged out by using a least squares fitting technique.

Results on several image sets are presented below. The first set is the Pentagon images which have been used to show steps of the registration methods in the previous sections. A set of images from the mars rover are used for the second set of images. Images taken over a mountain range show the registration method applied to a terrain scene. The fourth set of images is taken overlooking a Canadian city, showing buildings in both the foreground and in the background. Images of a three walled cube where each face is marked are used for the fifth set of images. A volcanic crater scene is used for the sixth set of images. Finally images from a flyover of the Ohio State University (OSU) campus are used in the sixth set.

6.1 Pentagon Images

These images show the projective geometric distortion caused when taking images of a scene from different points of view. The building is a large rigid object with the top of the building at one elevation and the background at another elevation. Planes at different elevations will have different projective transformations in the two images. This causes the projective transformation for the top of the building and the ground to be different. Figure 6.1 and Figure 6.2 show the two images, and the following figures show the overlaying and absolute difference images of each of the registration methods described above.

Figure 6.1 : Pentagon 1 image

Figure 6.2 : Pentagon 2 image

Figure 6.3 : Projective registration: Overlaying result

Figure 6.4 : Projective registration: Absolute difference result

Figure 6.5 : Piecewise linear registration: Overlaying result

Figure 6.6 : Piecewise linear registration: Absolute difference result

Figure 6.7 : Weighted linear registration: Overlaying result

Figure 6.8 : Weighted linear registration: Absolute difference result

Weighted linear registration attempts to produce a smooth mapping between the two images. While this improves some parts of the registration other parts are degraded due to the smoothing. Some areas along the edge of the building are degraded because there is a sharp change. These also exists errors along the left hand side of the building corresponding points fall in the shadow area. These points are a problem because the shadow is related to the height of the building and the position of the sun; it follows neither the ground nor the top of the building.

6.2 Mars Images

These images are taken from one of the mars rovers. Using the piecewise and weighted linear methods does not provide an improvement for this image set. The scene content is very regular and does not provide many ridged edges where corners can be located. This causes the piecewise method to have difficulty in locating stable points which can be triangulated. Due to the errors in the piecewise method the weighted linear result also suffers. However, the projective registration method is able to produce good results because this scene can be approximated by a single plane.

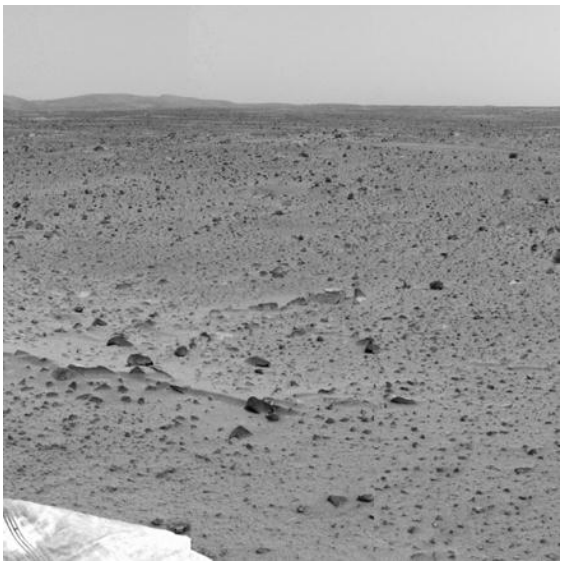

Figure 6.9: Mars 1

Figure 6.10 : Mars 2

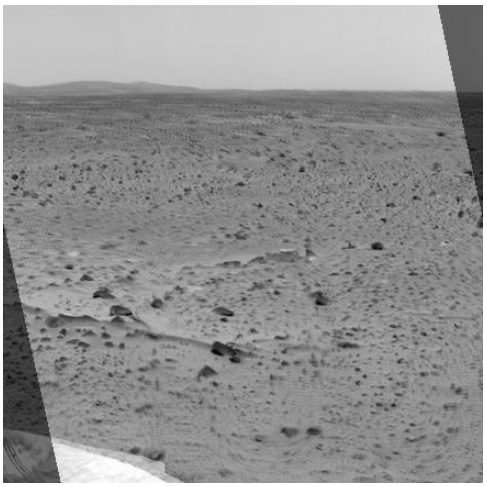

Figure 6.11 : Projective registration : Overlaying result

Figure 6.12 : Projective registration : Absolute difference result

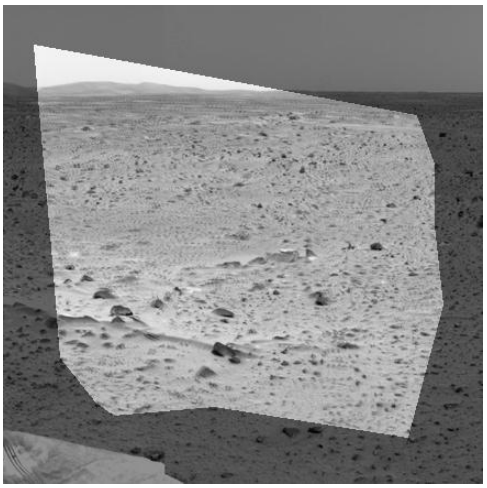

Figure 6.13 : Piecewise linear registration : Overlaying result

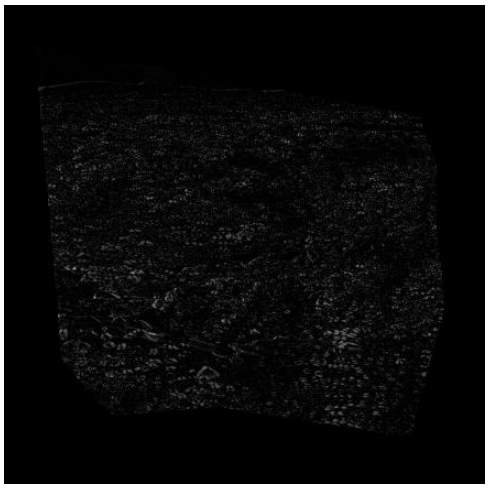

Figure 6.14 : Piecewise linear registration : Absolute difference result

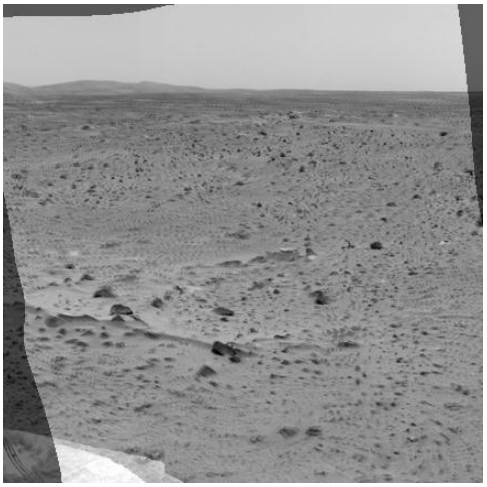

Figure 6.15 : Weighted linear registration : Overlaying result

Figure 6.16 : Weighted linear registration : Absolute difference result

6.3 Aerial Images

In another data set, images of a mountain range were used. Each registration method is applied and the results are shown below in the same fashion as before. This data set demonstrates the ability of the methods to register terrain images. Terrain images typically have surfaces with depth continuity rather than images of buildings and man-made structures that have discontinuous depth values and contain occlusions. The results obtained show that the weighted linear registration is able to account for variations in geometries of terrain images.

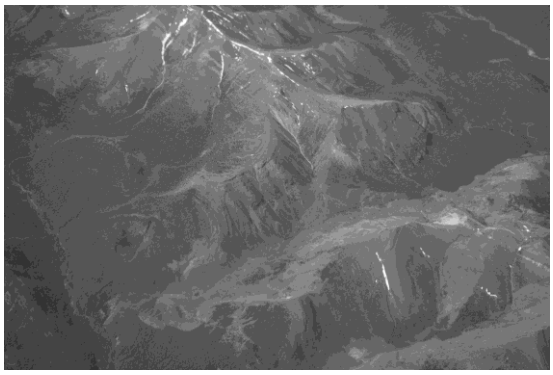

Figure 6.17 : Aerial 1 image

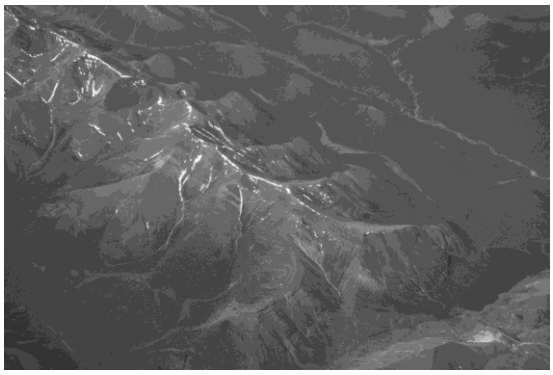

Figure 6.18 : Aerial 2 image

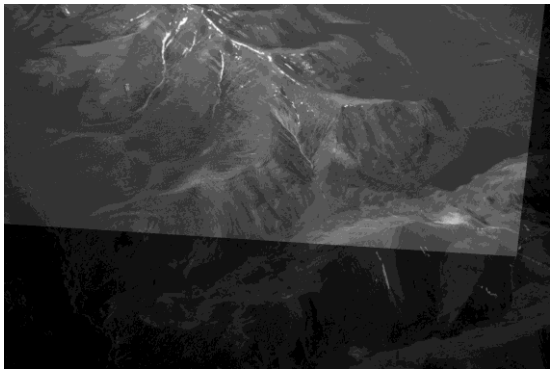

Figure 6.19 : Projective overlaying result

Figure 6.20 : Projective difference result

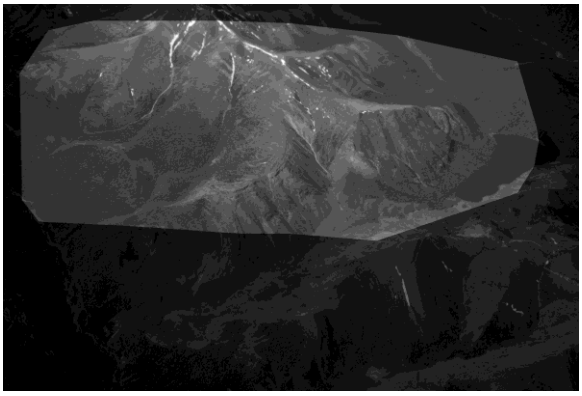

Figure 6.21 : Piecewise linear overlaying result

Figure 6.22 : Piecewise linear difference result

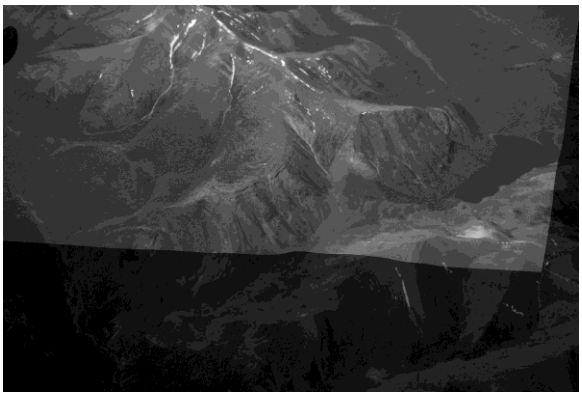

Figure 6.23 : Weighted linear overlaying result

Figure 6.24 : Weighted linear difference result

In the final result of the weighted linear registration method the lower right section of the image is poorly registered because there are no corresponding points in the region. It can be seen that this section is where there is a rise in the terrain starting another mountain range. In the upper right of the result, there are also errors because there are no corresponding points to compensate for changes in elevation.

6.4 City Images

This data set is interesting because of the buildings at different depths. There are a number of 2D planes which can be registered in this image, such as the face of the buildings or the road. Projective registration is only able to register one of the planes; in this case it registers the ground. The piecewise registration, by creating different planes, one for each triangle, is able to register the images much better.

Figure 6.25 : City 1 image

Figure 6.26 : City 2 image

Figure 6.27 : Projective overlaying result

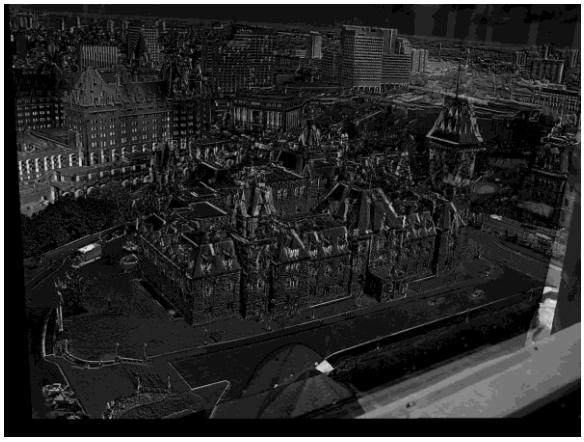

Figure 6.28 : Projective difference result

Figure 6.29 : Piecewise linear overlaying result

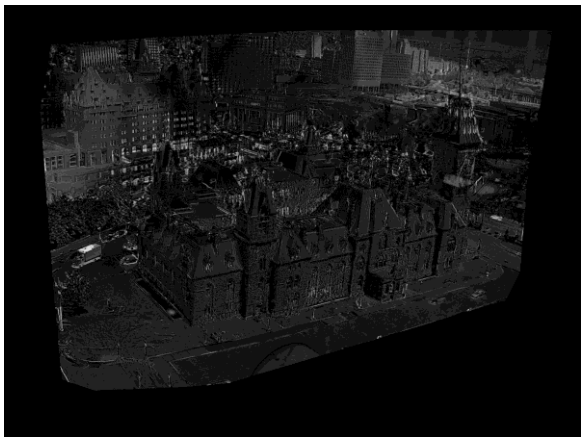

Figure 6.30 : Piecewise linear difference result

Figure 6.31 : Weighted linear overlaying result

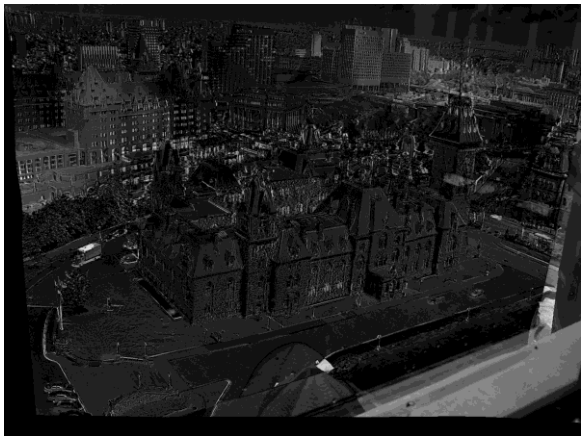

Figure 6.32 : Weighted linear difference result

Weighted linear registration demonstrates the ability to extend this registration beyond the convex hull of the points. An effect of the weighted linear registration is to smooth the transition between triangles. When a triangle edge lies on the edge of an actual plane, such as a transition between two faces of a building, the transition is smoothed causing registration inaccuracy.

6.5 Cube Images

These images demonstrate the difficulty of registering an image where projective difference between images varies from region to region. The sides of the cube are perpendicular to each other. Each side is labeled with numbers and X's that produce sufficient corner points to find local geometric difference between images. The projective registration method is only able to register one of the faces of the cube leaving large errors everywhere else. Even though the piecewise result is not able to find a registration for a large portion of the image it is able to register major portions of each face.

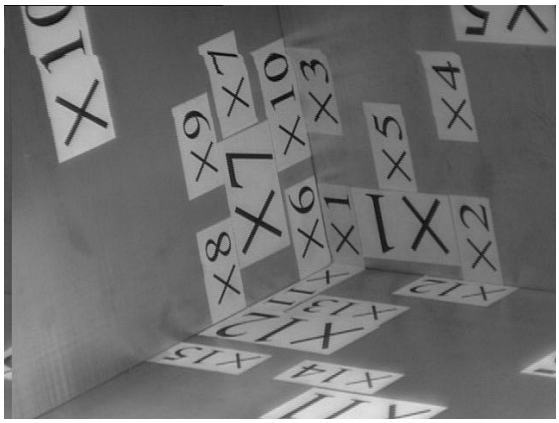

Figure 6.33 : Cube 1 image

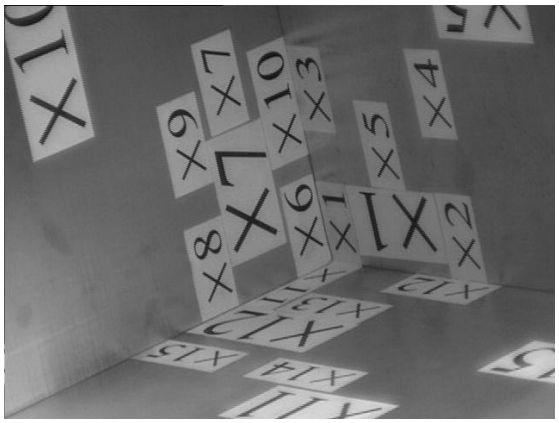

Figure 6.34 : Cube 2 image

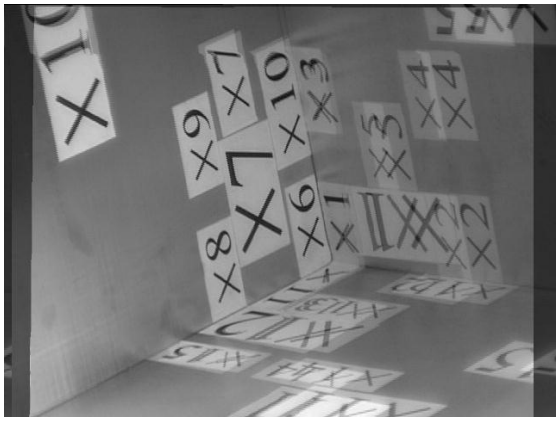

Figure 6.35 : Projective overlaying result

Figure 6.36 : Projective difference result

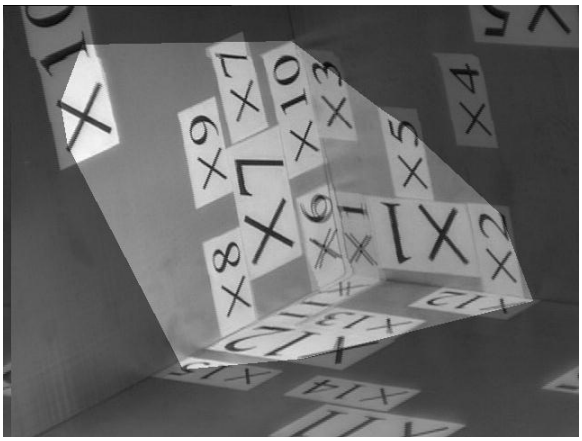

Figure 6.37 : Piecewise linear overlaying result

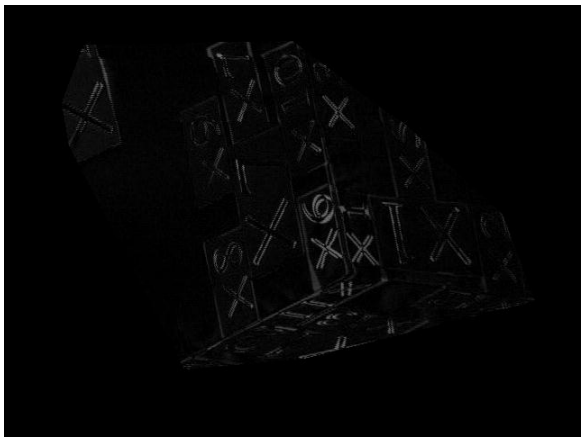

Figure 6.38 : Piecewise linear difference result

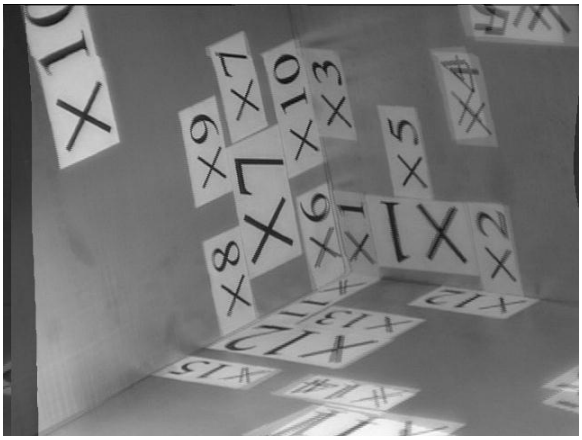

Figure 6.39 : Weighted linear overlaying result

Figure 6.40 : Weighted linear difference result

The weighted linear registration shows the ability of the method to extend the registration to the edges of the image domain. It can be seen that there are some areas where the registration has been degraded rather than improved. This is because of the smoothing from triangle to triangle where adjacent triangles lie in different planes in 3-D. There are significant errors in the upper right and bottom of the image. In these areas the Gaussian weights become weak allowing weightings from unrelated triangles to affect the mapping function in these regions.

6.6 Crater Images

This is a scene from the inside of a volcanic crater. In the background is the ridge of the crater. In this scene the foreground is relatively flat providing a good projective plane for registration. Projective registration performs well but still in unable to register the background ridge. Piecewise linear registration and weighted linear registration are able to produce results that more closely follow the surfaces in the scene.

Figure 6.41 : Crater 1 image

Figure 6.42 : Crater 2 image

Figure 6.43 : Projective overlaying result

Figure 6.44 : Projective difference result

Figure 6.45 : Piecewise linear overlaying result

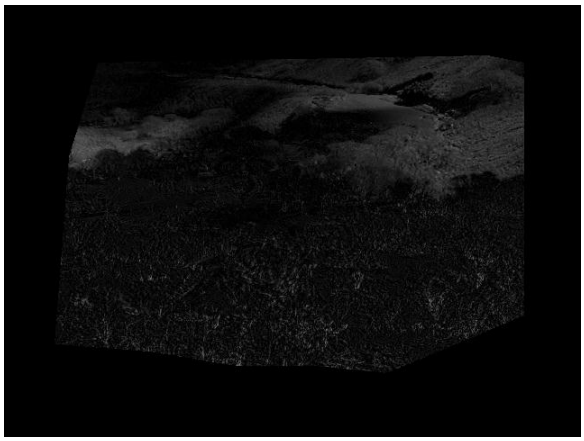

Figure 6.46 : Piecewise linear difference result

Figure 6.47 : Weighted linear overlaying result

Figure 6.48 : Weighted linear difference result

Weighted linear registration produces a significant improvement over the projective registration and improves the piecewise registration by extending the mapping over the whole image domain. There remains some error along the top of the ridge in the background; again this is caused by a lack of corresponding points along the ridge. The greatest contributor to the difference image is the change in cloud cover and scene lighting.

6.7 OSU Images

These images are taken by an unmanned aerial vehicle (UAV) over the Ohio State University campus. The projective difference between the images is not large as can be seen in the result of the projective registration. However, the stadium is a tall structure that produces geometric differences in the images, which cannot be compensated for. Piecewise registration shows an improvement by registering the images more accurately. In this case the weighted linear creates a significant error in registration around the stadium due to smoothing of the transformation near the stadium.

Figure 6.49 : OSU 1 image

Figure 6.50 : OSU 2 image

Figure 6.51 : Projective overlaying result

Figure 6.52 : Projective difference result

Figure 6.53 : Piecewise linear overlaying result

Figure 6.54 : Piecewise linear difference result

Figure 6.55 : Weighted linear overlaying result

Figure 6.56 : Weighted linear difference result

Chapter 7
Conclusion

7.1 Registration

Image registration is a computational method for bringing two images of a scene into the same coordinate system. There may be a need to compare the images or stitch them to create a panoramic mosaic of the scene. Methods for registering images using image intensities and image features exist in the literature. Gradient descent is a common method for registering images when using image intensities to register them. The alternative is to find similar control points in images and register images using corresponding control points. Locating control points and determining the correspondence between them can be a difficult task.

Many methods of image registration use the projective model. Projective models are only capable of aligning different views of a single plane. When registering views of a 3-D scene a single projective transformation is not sufficient to register the image. This study focused on this specific problem by exploring piecewise registration and weighted linear registration as a possible solution. Using the piecewise idea, a scene can be divided into smaller pieces

each of which can be approximated by a plane and registered by a projective or affine transformation.

7.2 Control-point detection

Locating control points in an image is an important step in image registration when using a control-point method. There exists a number of different methods for detecting control points in an image, each method is unique and captures different information from an image. Lowe [11] developed the Shift Invariant Feature Transform (SIFT) detector that produces scale and rotation invariant points in an image. Harris and Stephens [2] developed the well-known Harris corner detector. Each detector has its strengths and weaknesses. SIFT points are computationally intensive and Harris points are significantly simpler to compute. Depending on the application one method or another may be preferred. For this study the Harris corner detector was used as the interest point detector for its computational efficiency.

7.3 Piecewise linear registration

Piecewise linear registration proved to produce encouraging results in images with nonlinear geometric differences. The flexibility of the method to allow local geometric differences in images is significantly better than a strictly projective registration method. Piecewise linear, however; is sensitive to control point errors. If the program incorrectly determines point correspondences, then

there may be significant errors in the final mapping function. Locating correct corresponding points becomes significantly important for this method in order to produce accurate results. A significant amount of effort was put into accurately tracking the location of corresponding points as they travel from one scale to another.

7.4 Weighted linear registration

Weighted linear registration is an extension of the piecewise linear method. Each component of a mapping function maps the X or the Y coordinate from the reference image to the sensed image. In the piecewise approach the mapping function is not smooth. Each triangle creates an affine mapping function which is only valid over a specific triangle. To create a mapping function which is smooth and exists over the whole image domain the weighted linear method uses a rational Gaussian basis function to smooth the surface. A rational Gaussian weight function is centered at each triangle and is then averaged with neighboring weight functions to smoothly merge the mapping functions, creating a single smooth mapping function.

One of the issues with this method is when an edge of a triangle lies along a true edge of a planar region in the image. This edge will be smoothly merged with the mapping functions of adjacent triangles. This produces a mapping function that will no longer map the edge of the planar region. When considering images of buildings this may become an issue. The likelihood that this error will

be present in the results is increased when using a corner detector to locate interest points because, corners are at gradient intersections. This makes it highly probable that a triangle will be formed where one edge will be coincident with the edge of the object. Therefore, when registering images of buildings, actually piecewise linear is preferred over weighted linear.

Chapter 8
Future Work

This work demonstrated the ability of the piecewise linear and weighted linear methods to compensate for geometric differences between images of 3D scenes. Small errors in corresponding points can produce significant errors in the final registration. Two areas that require more work are the ability to find better corresponding points and the ability to find more corresponding points.

While corresponding points found in the lowest scale image are typically very accurate, the points tend to diverge as they are tracked through increasing scale. Locating the points accurately in higher scales is critically important for the method to be accurate. In this study, a template matching method was used to produce accurate results. Because template matching is not guaranteed to correctly locate corresponding points in the images, detected points may contain contaminated corresponding points. This method is forced to reject points when the templates do not produce high similarity. Even though this method proved to be effective in the type of images tested, it may suffer from throwing away points that may be critical in accurate registration of the images. Methods are needed that can retain more of the detected points.

In the developed method points were located at the lowest scale images and then tracked at successively higher scales. As the points are followed through higher scales the points become more evenly distributed in the image domain. This results in large spacing between the points in some image areas. As the number of levels is increased, the number of correspondences obtained decreases, increasing the spacing between some points. It will be more desirable to have points that fill all areas of an image.

Bibliography

[1] Zitova, B., & Flusser, J. Image Registration methods: a survey. *Image and Vision Computing , 21*, 977-1000. (2003).

[2] Harris, C., & Stephens, M. A combined corner and edge dtector. *Proceedings of the 4th Alvey Vision Conference* , 147-151. (1988).

[3] Lindeburg, T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision , 11* (3), 283-318. (1993).

[4] Fischler, M. A., & Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM , 24* (6), 381-395. (1981).

[5] Lindeburg, T. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics , 21* (2), 224-270. (1994).

[6] Lowe, D. G. Object recognition from local scale-invariant features. *International Conference on Computer Vision* , 1150-1157. (1999, September).

[7] Marr D., H. E. Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences , 207* (1167), 187-217. (1980).

[8] Moravec, H. P. Towards automatic visual obstacle avoidance. *Proceedings of the 5th International Joint Conference on Artificial Intelligence* , 584. (1977, August).

[9] Schmid, C., Mohr, R., & Bauckhage, C. Comparing and Evaluating Interest Points. *International Conference on Computer Vision* , 230-235. (1998).

[10] Wilson, H., & Giese, S. Theshold visibility of frequency gradient patterns. *Vision Res , 17*, 1177-1190. (1977).

[11] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision , 60* (2), 91-110. (2004).

[12] Goshtasby, A. Piecewise Linear Mapping Functions for Image Registration. *Pattern Recognition , 9* (6), 459-466. (1986).

[13] Goshtasby, A. "Registration of multi-view images". In LeMoigne, Netanyahoo, & Eastman (Eds.), *Image Registration for Remote Sensing* (pp. 153-178). Cambridge Press. (2011).

[14] Goshtasby, A. Piecewise Cubic Mapping Functions for Image Registration. *Pattern Recognition , 20* (5), 525-533. (1987).

[15] Goshtasby, A. *2-D and 3-D Image Registration for medical, remote sensing, and industrial applications.* Wiley-Interscience publication. (2005).

[16] Hartley, R., & Zisserman, A. *Multiple View Geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press. (2010).

[17] Burt, P. J. Fast filter transforms for image processing. *Computer Vision, Graphics, and Image Processing , 16*, pp. 20-51. (1981).

[18] Breg, M. d., Cheong, O., Kreveld, M. v., & Overmars, M. *Computational Geometry Algorithms and Applications* (3rd ed.). Berlin Heidelberg: Springer. (2008).

[19] Lindeburg, T. Scale-space for discrete signals. *IEEE Trans. Pattern Analysis and Machine Intell. , 12*, 234-254. (1990).

[20] Goshtasby, A. Registration of Images with Geometric Distortions. *IEEE Transactions on Geoscience and Remote Sensing , 26* (1), 60-64. (1988).

[21] Crowley, J. L. *A Representation for Visual Information.* PhD thesis, Carnegie-Mellon University, Robotics Insitute, Pittsburgh, Pennsylvania. (1981).

[22] Rohr, K. Modelling and Identification of Characteristic Intensity Variations". *Image and Vision Computing , 10* (2), 66-76. (1992).